### 11.1. Метод дифференциальной эволюции

Программно реализуйте метод дифференциальной эволюции и найдите с его помощью минимум заданной функции для всех указанных значений размерности $D$. Исследуйте влияние параметров на работу алгоритма. В отчет включить следующие данные:

1. Полную постановку задачи.
2. Исходный код программы.
3. График целевой функции для $D = 2$.
4. Наилучшее найденное оптимальное значение целевой функции и точка, в которой оно достигается.
5. Графики сходимости метода (по оси абсцисс — номер поколения, по оси ординат — текущее оптимальное значение погрешности, т. е. модуль разности точного и приближенного значения минимума). Ось ординат должна быть *логарифмической*! На одной плоскости изобразить вместе графики для трех «прогонов» алгоритма.
6. Среднее время работы алгоритма.
7. Для размерности $D = 2$ изобразить на плоскости линии уровня целевой функции и все точки популяции для четырех-шести различных поколений. Номера поколений выбрать таким образом, чтобы была видна сходимость метода.
8. Ответы на следующие вопросы:

   (a) Как зависит скорость сходимости от количества особей в популяции $N$?

   (b) Как зависит скорость сходимости от константы дифференцирования $F$?

   (c) Как зависит скорость сходимости от константы скрещивания $C$?

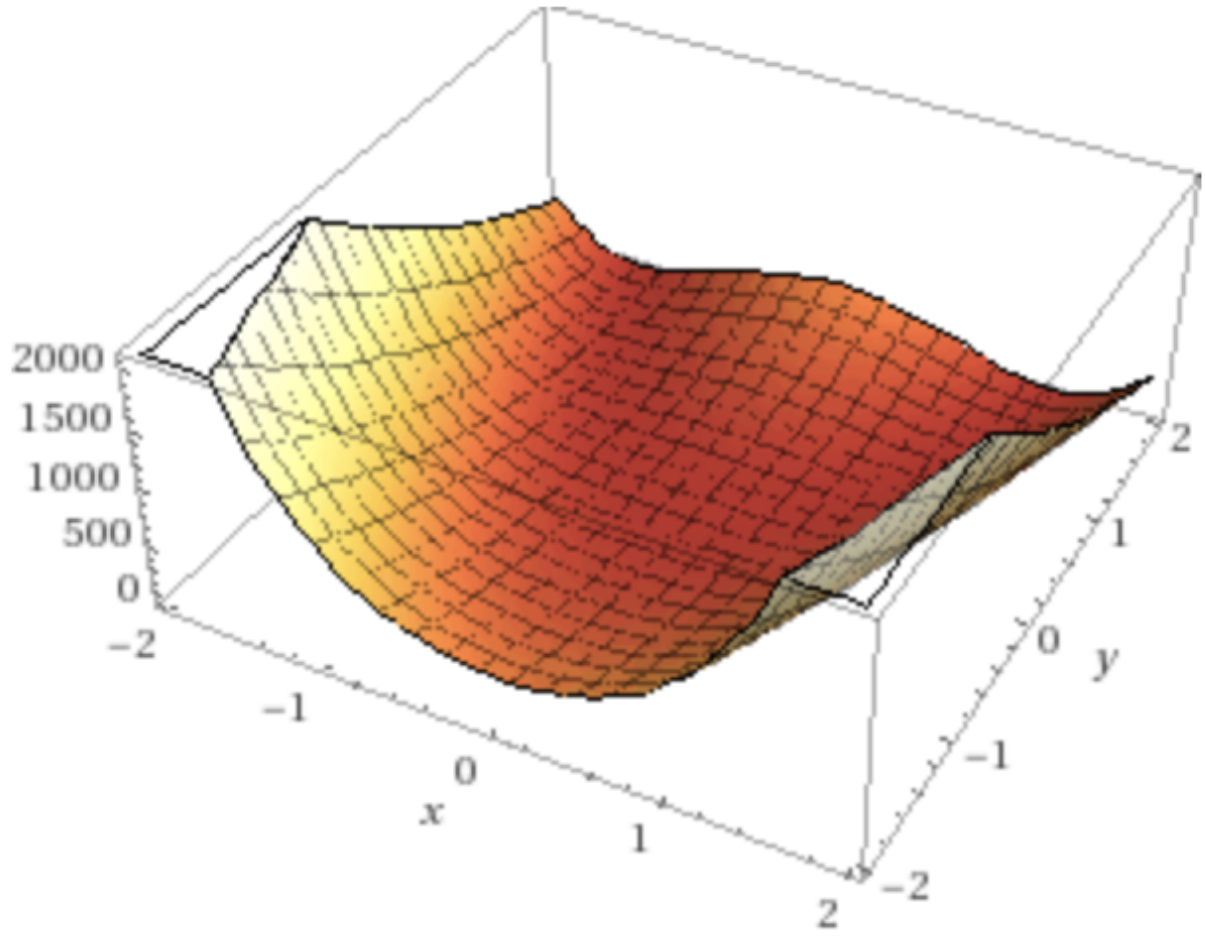
   (d) Какие значения параметров вам кажутся оптимальными?

   *Подтвердите свои выводы экспериментальными данными!*

**Варианты целевых функций**

**11.1.1.** Функция Розенброка. $f(x^*) = 0, \quad x^* = (1, 1, \ldots, 1)^T$.

$$f(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2), \quad x_i \in [-2, 2] \quad D = 2, 10, 100.$$


2000
1500
1000
500
0
−2
−1
0
1
2
x
2
1
0
−1
−2
y

In [1]:
```
import numpy as np
D = np.array([2, 10, 100], dtype=np.int)
limits = [-2.0, 2.0]
N_test = np.linspace(5, 25, 5, dtype=np.int)
C_test = np.linspace(0.1, 0.9, 5)
F_test = np.linspace(0.4, 1.3, 5)
```

In [30]:
```
def purpose_function(x):
    x_no_last = x[:-1]
    x_no_first = x[1:]
    return np.sum(100 * (x_no_last ** 2 - x_no_first) ** 2 + (1 - x_no_last) ** 2)


def purpose_function_population(population):
    N = population.shape[0]
    return np.min([purpose_function(population[i]) for i in range(N)])


def opt_plan(population):
    N = population.shape[0]
    opt_plan_index = np.argmax([purpose_function(population[i]) for i in range(N)])
    return population[opt_plan_index]


def get_population(D, N):
    result = np.random.uniform(low=limits[0], high=limits[1], size=(1,D))
    for i in range(N-1):
        item = np.random.uniform(low=limits[0], high=limits[1], size=(1,D))
        result = np.concatenate((result, item), axis=0)
    return result


def in_limits(x, limits):
    return np.all(np.logical_and(x <= limits[1], x >= limits[0]))
```

```python
def get_test_item(population, F, limits):
    size = population.shape[0]
    D = population.shape[1]
    indexes = np.random.randint(low=0, high=size, size=3)
    while(len(set(indexes)) != 3):
        indexes = np.random.randint(low=0, high=size, size=3)
    sample = population[indexes]
    beta = sample[2]
    delta = sample[1] - sample[0]
    omega = beta + F * delta
    if not in_limits(omega, limits):
        omega = get_population(D, 1).reshape((D, ))
    return omega


def cross(x, omega, C):
    mask = np.random.uniform(size=x.shape[0]) >= C
    return omega * mask + x * (1 - mask)


def evolution_step(population, F, C, limits):
    size = population.shape[0]
    for i in range(size):
        omega = get_test_item(population, F, limits)
        omega = cross(x=population[i], omega=omega, C=C)
        if purpose_function(omega) < purpose_function(population[i]):
            population[i] = omega


def full_process(D, N, F, C, limits, kNoBetter, log_populations=False):
    population = get_population(D, N)
    cur_purpose_function_value = purpose_function_population(population)
    log = [cur_purpose_function_value, ]
    log_populations = [population.copy(), ]
    cur_no_better = 0
    while cur_no_better < kNoBetter:
        evolution_step(population, F, C, limits)
        next_purpose_function_value = purpose_function_population(population)
        if cur_purpose_function_value <= next_purpose_function_value:
            cur_no_better += 1
        else:
            cur_purpose_function_value = next_purpose_function_value
            cur_no_better = 0
        if log_populations:
            log_populations.append(population.copy())
        log.append(cur_purpose_function_value)
        if len(log) > 10000:
            break
    return population, log, log_populations
```

```
def mega_test(D_test, N_test, F_test, C_test, limits, kNoBetter, log_populations=False, verbose=True):
    log = []
    for D in D_test:
        for N in N_test:
            for F in F_test:
                for C in C_test:
                    pop, logs, _ = full_process(D, N * D, F, C, limits, kNoBetter, log_populations)
                    log.append([D, N * D, F, C, opt_plan(pop), logs[-1], len(logs)])
                    if verbose:
                        print(log[-1])
    return log
```

In [7]:
```
def get_best_params(log):
    opt_value = np.min([log[i][5] for i in range(len(log))])
    opt_indexes = [i for i in range(len(log)) if log[i][5] == opt_value]
    min_iter_count = np.min([log[i][6] * log[i][1] for i in opt_indexes])
    index = [i for i in opt_indexes if log[i][6] * log[i][1] == min_iter_count][0]
    return log[index][:4]
```

In [8]:
```
log2 = log[:125]
log10 = log[125:250]
log100 = log[250:]
```

In [9]:
```
best_params = [get_best_params(log2), get_best_params(log10), get_best_params(log100)]
```

In [10]:
```
best_params
```

Out[10]:
```
[[2, 10, 0.8500000000000001, 0.1], [10, 50, 0.625, 0.1], [100, 2000, 0.4, 0.9]]
```

In [11]:
```
for i in range(len(log2)):
    print(log2[i][:4] + log2[i][5:])
```

```
[2, 10, 0.4, 0.1, 0.0075547205436353616, 245]
[2, 10, 0.4, 0.30000000000000004, 9.764079562813375e-20, 314]
[2, 10, 0.4, 0.5, 0.0, 498]
[2, 10, 0.4, 0.7000000000000001, 8.92944108682304e-09, 356]
[2, 10, 0.4, 0.9, 0.022742651209669028, 73]
[2, 10, 0.625, 0.1, 1.6951372325230164e-05, 984]
[2, 10, 0.625, 0.30000000000000004, 0.0, 345]
[2, 10, 0.625, 0.5, 0.061388573580402325, 8526]
[2, 10, 0.625, 0.7000000000000001, 0.006254521407091269, 81]
[2, 10, 0.625, 0.9, 0.2913996227490491, 79]
[2, 10, 0.8500000000000001, 0.1, 0.0, 285]
[2, 10, 0.8500000000000001, 0.30000000000000004, 0.0, 414]
[2, 10, 0.8500000000000001, 0.5, 1.232595164407831e-32, 762]
[2, 10, 0.8500000000000001, 0.7000000000000001, 0.029171099991794285, 98]
[2, 10, 0.8500000000000001, 0.9, 0.44314202555570986, 171]
[2, 10, 1.0750000000000002, 0.1, 0.0, 393]
[2, 10, 1.0750000000000002, 0.30000000000000004, 0.0, 581]
[2, 10, 1.0750000000000002, 0.5, 6.224605580259546e-28, 972]
[2, 10, 1.0750000000000002, 0.7000000000000001, 0.16720127769551882, 77]
[2, 10, 1.0750000000000002, 0.9, 1.8985728576691405, 51]
[2, 10, 1.3, 0.1, 1.232595164407831e-32, 455]
[2, 10, 1.3, 0.30000000000000004, 0.0003320095945635832, 101]
[2, 10, 1.3, 0.5, 0.0061410685323331085, 108]
[2, 10, 1.3, 0.7000000000000001, 0.0008775405641815477, 118]
[2, 10, 1.3, 0.9, 0.05135392249347848, 108]
[2, 20, 0.4, 0.1, 0.11740981037669847, 642]
[2, 20, 0.4, 0.30000000000000004, 0.0, 290]
[2, 20, 0.4, 0.5, 0.0015732368343850955, 66]
[2, 20, 0.4, 0.7000000000000001, 0.0010913415668850823, 70]
[2, 20, 0.4, 0.9, 0.03655043902839814, 137]
[2, 20, 0.625, 0.1, 0.0, 218]
[2, 20, 0.625, 0.30000000000000004, 0.0, 351]
[2, 20, 0.625, 0.5, 0.0, 651]
[2, 20, 0.625, 0.7000000000000001, 0.003560795378069257, 130]
[2, 20, 0.625, 0.9, 0.05732657432980966, 118]
[2, 20, 0.8500000000000001, 0.1, 0.0, 282]
[2, 20, 0.8500000000000001, 0.30000000000000004, 0.0, 441]
[2, 20, 0.8500000000000001, 0.5, 0.0, 783]
[2, 20, 0.8500000000000001, 0.7000000000000001, 0.027650114075060975, 51]
[2, 20, 0.8500000000000001, 0.9, 0.35576519867406625, 59]
[2, 20, 1.0750000000000002, 0.1, 0.0, 363]
[2, 20, 1.0750000000000002, 0.30000000000000004, 0.0, 497]
[2, 20, 1.0750000000000002, 0.5, 7.888609052210118e-31, 773]
[2, 20, 1.0750000000000002, 0.7000000000000001, 0.0064983560659206165, 139]
[2, 20, 1.0750000000000002, 0.9, 0.03356306230162563, 104]
```

[2, 20, 1.3, 0.1, 0.0, 492]
[2, 20, 1.3, 0.30000000000000004, 0.0, 690]
[2, 20, 1.3, 0.5, 0.0001345773084148679, 184]
[2, 20, 1.3, 0.7000000000000001, 0.001027915749054359, 179]
[2, 20, 1.3, 0.9, 0.036281638922595796, 83]
[2, 30, 0.4, 0.1, 0.00210324644107139, 381]
[2, 30, 0.4, 0.30000000000000004, 0.0, 289]
[2, 30, 0.4, 0.5, 0.0, 462]
[2, 30, 0.4, 0.7000000000000001, 0.00015995768988887396, 136]
[2, 30, 0.4, 0.9, 0.00267104969226578, 122]
[2, 30, 0.625, 0.1, 0.0, 226]
[2, 30, 0.625, 0.30000000000000004, 0.0, 356]
[2, 30, 0.625, 0.5, 0.0, 629]
[2, 30, 0.625, 0.7000000000000001, 0.038718027434234935, 82]
[2, 30, 0.625, 0.9, 0.0945940729531096, 159]
[2, 30, 0.8500000000000001, 0.1, 0.0, 290]
[2, 30, 0.8500000000000001, 0.30000000000000004, 4.871709127805511e-28, 310]
[2, 30, 0.8500000000000001, 0.5, 0.0, 761]
[2, 30, 0.8500000000000001, 0.7000000000000001, 0.001185968202618958, 121]
[2, 30, 0.8500000000000001, 0.9, 0.022004689274563255, 120]
[2, 30, 1.0750000000000002, 0.1, 0.0, 365]
[2, 30, 1.0750000000000002, 0.30000000000000004, 1.8700707964881117e-10, 156]
[2, 30, 1.0750000000000002, 0.5, 0.007914814117929667, 92]
[2, 30, 1.0750000000000002, 0.7000000000000001, 0.013350771723861381, 124]
[2, 30, 1.0750000000000002, 0.9, 0.003924630045735877, 76]
[2, 30, 1.3, 0.1, 0.0, 472]
[2, 30, 1.3, 0.30000000000000004, 0.0, 774]
[2, 30, 1.3, 0.5, 0.0038423709755555516, 109]
[2, 30, 1.3, 0.7000000000000001, 0.004519179911435774, 94]
[2, 30, 1.3, 0.9, 0.0956767145454532, 102]
[2, 40, 0.4, 0.1, 0.0, 178]
[2, 40, 0.4, 0.30000000000000004, 0.0, 272]
[2, 40, 0.4, 0.5, 0.0, 492]
[2, 40, 0.4, 0.7000000000000001, 0.004604396587087151, 90]
[2, 40, 0.4, 0.9, 0.011778821391234831, 80]
[2, 40, 0.625, 0.1, 0.0, 220]
[2, 40, 0.625, 0.30000000000000004, 0.0, 337]
[2, 40, 0.625, 0.5, 5.424853327191605e-16, 263]
[2, 40, 0.625, 0.7000000000000001, 1.3339822447008237e-06, 277]
[2, 40, 0.625, 0.9, 0.027539057712026874, 322]
[2, 40, 0.8500000000000001, 0.1, 0.0, 292]
[2, 40, 0.8500000000000001, 0.30000000000000004, 0.0, 428]
[2, 40, 0.8500000000000001, 0.5, 0.0, 750]
[2, 40, 0.8500000000000001, 0.7000000000000001, 0.0009679646747280349, 182]
[2, 40, 0.8500000000000001, 0.9, 0.07457881515307646, 70]

```
[2, 40, 1.0750000000000002, 0.1, 0.0, 349]
[2, 40, 1.0750000000000002, 0.30000000000000004, 0.0, 545]
[2, 40, 1.0750000000000002, 0.5, 0.0, 1032]
[2, 40, 1.0750000000000002, 0.7000000000000001, 9.328518103924046e-05, 241]
[2, 40, 1.0750000000000002, 0.9, 0.06206081666800094, 90]
[2, 40, 1.3, 0.1, 0.0, 450]
[2, 40, 1.3, 0.30000000000000004, 0.006260294448380858, 94]
[2, 40, 1.3, 0.5, 0.0004959531968819704, 132]
[2, 40, 1.3, 0.7000000000000001, 0.010666357787357951, 58]
[2, 40, 1.3, 0.9, 0.14061093751565962, 62]
[2, 50, 0.4, 0.1, 0.0, 171]
[2, 50, 0.4, 0.30000000000000004, 0.0, 264]
[2, 50, 0.4, 0.5, 0.0, 471]
[2, 50, 0.4, 0.7000000000000001, 4.187380674173401e-23, 784]
[2, 50, 0.4, 0.9, 0.005879301686721076, 68]
[2, 50, 0.625, 0.1, 0.0, 228]
[2, 50, 0.625, 0.30000000000000004, 0.0, 340]
[2, 50, 0.625, 0.5, 0.0, 632]
[2, 50, 0.625, 0.7000000000000001, 0.011953223719995562, 80]
[2, 50, 0.625, 0.9, 0.016343151249259353, 53]
[2, 50, 0.8500000000000001, 0.1, 0.0, 289]
[2, 50, 0.8500000000000001, 0.30000000000000004, 0.0, 438]
[2, 50, 0.8500000000000001, 0.5, 1.7542485253951398e-15, 337]
[2, 50, 0.8500000000000001, 0.7000000000000001, 0.003962810232465607, 86]
[2, 50, 0.8500000000000001, 0.9, 0.032524207219607575, 97]
[2, 50, 1.0750000000000002, 0.1, 0.0, 352]
[2, 50, 1.0750000000000002, 0.30000000000000004, 0.0, 541]
[2, 50, 1.0750000000000002, 0.5, 0.0, 966]
[2, 50, 1.0750000000000002, 0.7000000000000001, 0.0013280253825149059, 125]
[2, 50, 1.0750000000000002, 0.9, 0.0843869673283688, 53]
[2, 50, 1.3, 0.1, 0.0, 432]
[2, 50, 1.3, 0.30000000000000004, 3.178808819260786e-13, 241]
[2, 50, 1.3, 0.5, 2.4775685659983928e-05, 157]
[2, 50, 1.3, 0.7000000000000001, 0.0005169356683849733, 168]
[2, 50, 1.3, 0.9, 0.027774661585020108, 102]
```

[10, 50, 0.4, 0.1, 7.125524745831758, 10001]
[10, 50, 0.4, 0.30000000000000004, 6.939210927523591, 10001]
[10, 50, 0.4, 0.5, 4.575533031588241, 10001]
[10, 50, 0.4, 0.7000000000000001, 4.156846480731334, 10001]
[10, 50, 0.4, 0.9, 4.264742018459562, 514]
[10, 50, 0.625, 0.1, 0.0, 2033]
[10, 50, 0.625, 0.30000000000000004, 0.00014226302143774186, 1117]
[10, 50, 0.625, 0.5, 0.6300307184828351, 1367]
[10, 50, 0.625, 0.7000000000000001, 5.704446910993907, 323]
[10, 50, 0.625, 0.9, 4.759500283354953, 440]
[10, 50, 0.8500000000000001, 0.1, 0.00418002707113865, 1017]
[10, 50, 0.8500000000000001, 0.30000000000000004, 33.096328368621954, 124]
[10, 50, 0.8500000000000001, 0.5, 4.796945040162304, 511]
[10, 50, 0.8500000000000001, 0.7000000000000001, 4.859142232913869, 478]
[10, 50, 0.8500000000000001, 0.9, 4.141552550332722, 453]
[10, 50, 1.0750000000000002, 0.1, 157.95541709028913, 92]
[10, 50, 1.0750000000000002, 0.30000000000000004, 51.824955342996816, 254]
[10, 50, 1.0750000000000002, 0.5, 6.469608372029613, 443]
[10, 50, 1.0750000000000002, 0.7000000000000001, 8.993342443237482, 391]
[10, 50, 1.0750000000000002, 0.9, 8.089567654586478, 290]
[10, 50, 1.3, 0.1, 132.46871214446537, 133]
[10, 50, 1.3, 0.30000000000000004, 180.18306387245408, 69]
[10, 50, 1.3, 0.5, 151.20203900791918, 86]
[10, 50, 1.3, 0.7000000000000001, 15.70704066180031, 267]
[10, 50, 1.3, 0.9, 20.746360528673634, 237]
[10, 100, 0.4, 0.1, 1.6684493586686753, 8354]
[10, 100, 0.4, 0.30000000000000004, 0.1824970238130808, 10001]
[10, 100, 0.4, 0.5, 0.15762075027299266, 10001]
[10, 100, 0.4, 0.7000000000000001, 4.575475419893386, 411]
[10, 100, 0.4, 0.9, 2.032010150691329, 425]
[10, 100, 0.625, 0.1, 0.0, 2221]
[10, 100, 0.625, 0.30000000000000004, 0.0016627627813528504, 936]
[10, 100, 0.625, 0.5, 0.0005885105320427923, 2565]
[10, 100, 0.625, 0.7000000000000001, 4.682895495534398, 561]
[10, 100, 0.625, 0.9, 5.518307885573752, 361]
[10, 100, 0.8500000000000001, 0.1, 0.32592374710401106, 751]
[10, 100, 0.8500000000000001, 0.30000000000000004, 0.029956382660915493, 1335]
[10, 100, 0.8500000000000001, 0.5, 2.8264247513364076, 1045]
[10, 100, 0.8500000000000001, 0.7000000000000001, 5.837539804481543, 357]
[10, 100, 0.8500000000000001, 0.9, 9.206564647857942, 288]
[10, 100, 1.0750000000000002, 0.1, 210.68340490597976, 60]
[10, 100, 1.0750000000000002, 0.30000000000000004, 75.9764097074882, 105]
[10, 100, 1.0750000000000002, 0.5, 20.177016295197696, 260]
[10, 100, 1.0750000000000002, 0.7000000000000001, 19.99958887599699, 179]
[10, 100, 1.0750000000000002, 0.9, 27.72707858357944, 146]

[10, 100, 1.3, 0.1, 155.10454697621805, 75]
[10, 100, 1.3, 0.30000000000000004, 71.61166479285961, 272]
[10, 100, 1.3, 0.5, 55.96065990071313, 130]
[10, 100, 1.3, 0.7000000000000001, 16.524166584590734, 242]
[10, 100, 1.3, 0.9, 9.012101488368192, 333]
[10, 150, 0.4, 0.1, 0.0012197883481595184, 10001]
[10, 150, 0.4, 0.30000000000000004, 0.02494747228427139, 10001]
[10, 150, 0.4, 0.5, 0.06477047438401741, 10001]
[10, 150, 0.4, 0.7000000000000001, 5.380811492382517, 570]
[10, 150, 0.4, 0.9, 5.2270033275127865, 379]
[10, 150, 0.625, 0.1, 0.0, 2354]
[10, 150, 0.625, 0.30000000000000004, 6.392602249491929e-07, 1520]
[10, 150, 0.625, 0.5, 0.002439897878568674, 2351]
[10, 150, 0.625, 0.7000000000000001, 4.865227011466315, 569]
[10, 150, 0.625, 0.9, 3.5690398981708866, 576]
[10, 150, 0.8500000000000001, 0.1, 0.0014044680855148028, 1278]
[10, 150, 0.8500000000000001, 0.30000000000000004, 0.8076868778257109, 928]
[10, 150, 0.8500000000000001, 0.5, 1.508126387702878, 1340]
[10, 150, 0.8500000000000001, 0.7000000000000001, 6.678699828707334, 340]
[10, 150, 0.8500000000000001, 0.9, 6.288559297256784, 272]
[10, 150, 1.0750000000000002, 0.1, 114.06531901414859, 88]
[10, 150, 1.0750000000000002, 0.30000000000000004, 66.3666415302729, 151]
[10, 150, 1.0750000000000002, 0.5, 25.50457770623109, 185]
[10, 150, 1.0750000000000002, 0.7000000000000001, 7.697336140728796, 399]
[10, 150, 1.0750000000000002, 0.9, 4.916859770934079, 476]
[10, 150, 1.3, 0.1, 89.76998941773223, 97]
[10, 150, 1.3, 0.30000000000000004, 93.83247217088227, 134]
[10, 150, 1.3, 0.5, 38.04207852208573, 247]
[10, 150, 1.3, 0.7000000000000001, 21.60648501573974, 183]
[10, 150, 1.3, 0.9, 5.896373212657905, 324]
[10, 200, 0.4, 0.1, 0.0, 1175]
[10, 200, 0.4, 0.30000000000000004, 0.008466894674789071, 10001]
[10, 200, 0.4, 0.5, 0.052354954460231994, 10001]
[10, 200, 0.4, 0.7000000000000001, 5.344318676648666, 724]
[10, 200, 0.4, 0.9, 2.5996509015306852, 503]
[10, 200, 0.625, 0.1, 0.0, 2300]
[10, 200, 0.625, 0.30000000000000004, 0.0066829685343034835, 850]
[10, 200, 0.625, 0.5, 0.004014929030296958, 2204]
[10, 200, 0.625, 0.7000000000000001, 5.434094817933357, 646]
[10, 200, 0.625, 0.9, 5.004573632553461, 375]
[10, 200, 0.8500000000000001, 0.1, 10.550097865403343, 290]
[10, 200, 0.8500000000000001, 0.30000000000000004, 2.220767714053266, 726]
[10, 200, 0.8500000000000001, 0.5, 6.092240703478839, 369]
[10, 200, 0.8500000000000001, 0.7000000000000001, 5.300397655353441, 436]
[10, 200, 0.8500000000000001, 0.9, 7.326688272034919, 291]

```
[10, 200, 1.0750000000000002, 0.1, 71.98618358166733, 147]
[10, 200, 1.0750000000000002, 0.30000000000000004, 84.17991351277189, 64]
[10, 200, 1.0750000000000002, 0.5, 5.9628244343298435, 535]
[10, 200, 1.0750000000000002, 0.7000000000000001, 6.8539868047891135, 237]
[10, 200, 1.0750000000000002, 0.9, 9.755766866186796, 273]
lick to scroll output; double click to hide
                                                  172849526124925, 156]
[10, 200, 1.3, 0.30000000000000004, 61.86927145418933, 192]
[10, 200, 1.3, 0.5, 34.33642472010623, 149]
[10, 200, 1.3, 0.7000000000000001, 18.593020965618045, 173]
[10, 200, 1.3, 0.9, 4.846134552444858, 413]
[10, 250, 0.4, 0.1, 0.0, 1438]
[10, 250, 0.4, 0.30000000000000004, 0.005752206038446479, 10001]
[10, 250, 0.4, 0.5, 0.042284131937999604, 10001]
[10, 250, 0.4, 0.7000000000000001, 5.189710776106448, 308]
[10, 250, 0.4, 0.9, 8.701649577771551, 199]
[10, 250, 0.625, 0.1, 0.0, 2381]
[10, 250, 0.625, 0.30000000000000004, 0.0030622651483406377, 909]
[10, 250, 0.625, 0.5, 0.019346677837430257, 1992]
[10, 250, 0.625, 0.7000000000000001, 4.692279748117338, 358]
[10, 250, 0.625, 0.9, 7.39051847024082, 275]
[10, 250, 0.8500000000000001, 0.1, 0.009195183271860499, 1034]
[10, 250, 0.8500000000000001, 0.30000000000000004, 42.49069049111357, 113]
[10, 250, 0.8500000000000001, 0.5, 8.06508272225564, 220]
[10, 250, 0.8500000000000001, 0.7000000000000001, 4.386117307761033, 653]
[10, 250, 0.8500000000000001, 0.9, 14.681237059675686, 137]
[10, 250, 1.0750000000000002, 0.1, 120.12986581556187, 68]
[10, 250, 1.0750000000000002, 0.30000000000000004, 92.24425515944054, 64]
[10, 250, 1.0750000000000002, 0.5, 10.216241621724066, 274]
[10, 250, 1.0750000000000002, 0.7000000000000001, 6.0393302508529985, 419]
[10, 250, 1.0750000000000002, 0.9, 12.487803567170241, 210]
[10, 250, 1.3, 0.1, 98.85794303105078, 80]
[10, 250, 1.3, 0.30000000000000004, 34.29148273742467, 168]
[10, 250, 1.3, 0.5, 30.05200460005637, 214]
[10, 250, 1.3, 0.7000000000000001, 8.854760058234113, 351]
[10, 250, 1.3, 0.9, 12.439343545902693, 260]
```

```
[100, 500, 0.4, 0.1, 19262.241074164896, 111]
[100, 500, 0.4, 0.30000000000000004, 19145.152087826176, 133]
[100, 500, 0.4, 0.5, 16159.768933227058, 107]
[100, 500, 0.4, 0.7000000000000001, 12399.127887301105, 177]
[100, 500, 0.4, 0.9, 4357.814441982644, 491]
[100, 500, 0.625, 0.1, 21785.05569723643, 63]
[100, 500, 0.625, 0.30000000000000004, 20744.85375211873, 60]
[100, 500, 0.625, 0.5, 17903.201686337445, 169]
[100, 500, 0.625, 0.7000000000000001, 15134.97544546678, 120]
[100, 500, 0.625, 0.9, 6653.171507681255, 357]
[100, 500, 0.8500000000000001, 0.1, 22657.502279542507, 127]
[100, 500, 0.8500000000000001, 0.30000000000000004, 19402.27882680908, 115]
[100, 500, 0.8500000000000001, 0.5, 14579.368940196811, 166]
[100, 500, 0.8500000000000001, 0.7000000000000001, 14839.61066567215, 173]
[100, 500, 0.8500000000000001, 0.9, 8214.426230514791, 193]
[100, 500, 1.0750000000000002, 0.1, 21826.34136030758, 155]
[100, 500, 1.0750000000000002, 0.30000000000000004, 17490.85839420266, 138]
[100, 500, 1.0750000000000002, 0.5, 17724.59799632441, 138]
[100, 500, 1.0750000000000002, 0.7000000000000001, 13926.73381598356, 114]
[100, 500, 1.0750000000000002, 0.9, 7435.540008363014, 286]
[100, 500, 1.3, 0.1, 19945.503716748633, 96]
[100, 500, 1.3, 0.30000000000000004, 18193.888337379027, 122]
[100, 500, 1.3, 0.5, 17420.62907598465, 117]
[100, 500, 1.3, 0.7000000000000001, 14360.22219027117, 155]
[100, 500, 1.3, 0.9, 8386.505667994667, 241]
[100, 1000, 0.4, 0.1, 21228.146886216775, 69]
[100, 1000, 0.4, 0.30000000000000004, 18597.776764875496, 64]
[100, 1000, 0.4, 0.5, 17820.767970303834, 93]
[100, 1000, 0.4, 0.7000000000000001, 13449.626026708094, 180]
[100, 1000, 0.4, 0.9, 7511.725298671333, 228]
[100, 1000, 0.625, 0.1, 20726.07788835142, 82]
[100, 1000, 0.625, 0.30000000000000004, 18893.854075658528, 138]
[100, 1000, 0.625, 0.5, 16989.346604020295, 75]
[100, 1000, 0.625, 0.7000000000000001, 12280.224164349524, 260]
[100, 1000, 0.625, 0.9, 8216.583392531365, 207]
[100, 1000, 0.8500000000000001, 0.1, 18688.819976626477, 127]
[100, 1000, 0.8500000000000001, 0.30000000000000004, 16731.686943042117, 228]
[100, 1000, 0.8500000000000001, 0.5, 16437.42912534443, 123]
[100, 1000, 0.8500000000000001, 0.7000000000000001, 13611.682797124655, 167]
[100, 1000, 0.8500000000000001, 0.9, 6472.626346820909, 313]
[100, 1000, 1.0750000000000002, 0.1, 17929.494125385023, 264]
[100, 1000, 1.0750000000000002, 0.30000000000000004, 19310.237898370342, 76]
[100, 1000, 1.0750000000000002, 0.5, 15692.429530417963, 156]
[100, 1000, 1.0750000000000002, 0.7000000000000001, 11328.298634203837, 188]
[100, 1000, 1.0750000000000002, 0.9, 7185.704068130942, 263]
```

[100, 1000, 1.3, 0.1, 18739.979288788676, 213]
[100, 1000, 1.3, 0.30000000000000004, 19393.085289575884, 108]
[100, 1000, 1.3, 0.5, 17313.422084578066, 154]
[100, 1000, 1.3, 0.7000000000000001, 12751.292486860782, 182]
[100, 1000, 1.3, 0.9, 6080.898853107058, 341]
[100, 1500, 0.4, 0.1, 19575.118542471122, 192]
[100, 1500, 0.4, 0.30000000000000004, 15690.873431637614, 126]
[100, 1500, 0.4, 0.5, 16574.452909679883, 116]
[100, 1500, 0.4, 0.7000000000000001, 13436.361775808611, 211]
[100, 1500, 0.4, 0.9, 7666.220823857281, 237]
[100, 1500, 0.625, 0.1, 19977.339952273465, 68]
[100, 1500, 0.625, 0.30000000000000004, 17722.87096732506, 135]
[100, 1500, 0.625, 0.5, 16830.151271495994, 125]
[100, 1500, 0.625, 0.7000000000000001, 13870.241350115535, 167]
[100, 1500, 0.625, 0.9, 7256.355520784322, 263]
[100, 1500, 0.8500000000000001, 0.1, 20358.44195428803, 63]
[100, 1500, 0.8500000000000001, 0.30000000000000004, 18774.06656401645, 140]
[100, 1500, 0.8500000000000001, 0.5, 17692.779197081625, 76]
[100, 1500, 0.8500000000000001, 0.7000000000000001, 12244.600434994962, 206]
[100, 1500, 0.8500000000000001, 0.9, 5952.615663381747, 424]
[100, 1500, 1.0750000000000002, 0.1, 20277.628882552093, 119]
[100, 1500, 1.0750000000000002, 0.30000000000000004, 16694.03384761649, 189]
[100, 1500, 1.0750000000000002, 0.5, 16358.28032220993, 160]
[100, 1500, 1.0750000000000002, 0.7000000000000001, 14501.875704887347, 127]
[100, 1500, 1.0750000000000002, 0.9, 7024.3803410362125, 298]
[100, 1500, 1.3, 0.1, 20204.867107123035, 101]
[100, 1500, 1.3, 0.30000000000000004, 18312.43232553858, 136]
[100, 1500, 1.3, 0.5, 16754.766698517717, 85]
[100, 1500, 1.3, 0.7000000000000001, 13326.135862911451, 143]
[100, 1500, 1.3, 0.9, 7385.551372412606, 316]
[100, 2000, 0.4, 0.1, 18241.379256120796, 61]
[100, 2000, 0.4, 0.30000000000000004, 14412.608405840247, 133]
[100, 2000, 0.4, 0.5, 14437.75587693334, 326]
[100, 2000, 0.4, 0.7000000000000001, 15090.393581179402, 69]
[100, 2000, 0.4, 0.9, 93.78083363907132, 5820]
[100, 2000, 0.625, 0.1, 21399.58842826814, 96]
[100, 2000, 0.625, 0.30000000000000004, 19045.317811957797, 82]
[100, 2000, 0.625, 0.5, 16082.16604381351, 169]
[100, 2000, 0.625, 0.7000000000000001, 10647.957595520169, 230]
[100, 2000, 0.625, 0.9, 6647.80550097407, 361]
[100, 2000, 0.8500000000000001, 0.1, 20276.22267723384, 61]
[100, 2000, 0.8500000000000001, 0.30000000000000004, 18734.481731944725, 90]
[100, 2000, 0.8500000000000001, 0.5, 16199.594430561405, 166]
[100, 2000, 0.8500000000000001, 0.7000000000000001, 14756.591530500675, 96]
[100, 2000, 0.8500000000000001, 0.9, 5935.980489809146, 342]

[100, 2000, 1.0750000000000002, 0.1, 20140.890889726226, 78]
[100, 2000, 1.0750000000000002, 0.30000000000000004, 17188.639864378827, 93]
[100, 2000, 1.0750000000000002, 0.5, 15754.931213963593, 145]
[100, 2000, 1.0750000000000002, 0.7000000000000001, 12338.19697520998, 271]
[100, 2000, 1.0750000000000002, 0.9, 5698.752117310571, 445]
[100, 2000, 1.3, 0.1, 19832.22291718708, 57]
[100, 2000, 1.3, 0.30000000000000004, 19368.915145673214, 66]
[100, 2000, 1.3, 0.5, 17346.428157957056, 92]
[100, 2000, 1.3, 0.7000000000000001, 12561.545891781067, 163]
[100, 2000, 1.3, 0.9, 5511.986195260103, 469]
[100, 2500, 0.4, 0.1, 20825.540162328125, 95]
[100, 2500, 0.4, 0.30000000000000004, 17249.204918425632, 121]
[100, 2500, 0.4, 0.5, 15870.284615263707, 174]
[100, 2500, 0.4, 0.7000000000000001, 11253.1392236335, 271]
[100, 2500, 0.4, 0.9, 8287.157339261434, 184]
[100, 2500, 0.625, 0.1, 18107.291372630527, 62]
[100, 2500, 0.625, 0.30000000000000004, 18177.788533660034, 169]
[100, 2500, 0.625, 0.5, 16589.140117093186, 130]
[100, 2500, 0.625, 0.7000000000000001, 12759.14622882156, 143]
[100, 2500, 0.625, 0.9, 6962.20140682216, 290]
[100, 2500, 0.8500000000000001, 0.1, 20374.594556971282, 98]
[100, 2500, 0.8500000000000001, 0.30000000000000004, 18855.480331219816, 64]
[100, 2500, 0.8500000000000001, 0.5, 15643.458748334975, 174]
[100, 2500, 0.8500000000000001, 0.7000000000000001, 13105.396701817659, 138]
[100, 2500, 0.8500000000000001, 0.9, 6043.330429489843, 459]
[100, 2500, 1.0750000000000002, 0.1, 18677.059612399305, 114]
[100, 2500, 1.0750000000000002, 0.30000000000000004, 19075.82170885921, 72]
[100, 2500, 1.0750000000000002, 0.5, 14235.386778388127, 81]
[100, 2500, 1.0750000000000002, 0.7000000000000001, 12315.185469198566, 228]
[100, 2500, 1.0750000000000002, 0.9, 6910.649956389017, 302]
[100, 2500, 1.3, 0.1, 19604.180118209795, 71]
[100, 2500, 1.3, 0.30000000000000004, 16454.01734790696, 120]
[100, 2500, 1.3, 0.5, 16369.029110692822, 121]
[100, 2500, 1.3, 0.7000000000000001, 11279.940433358124, 223]
[100, 2500, 1.3, 0.9, 7102.773317668086, 291]

```
import time
kTimes = 10
start = time.time()
for i in range(kTimes):
    bp_proc2, bp_log2, _ = full_process(best_params[0][0], best_params[0][1], best_params[0][2],
                                        best_params[0][3], limits, 50)
print('D=2 mean work time: {} seconds.'.format((time.time() - start) / kTimes))
```

```
D=2 mean work time: 0.19593801498413085 seconds.
```

```
start = time.time()
kTimes = 5
for i in range(kTimes):
    bp_proc10, bp_log10, _ = full_process(best_params[1][0], best_params[1][1], best_params[1][2],
                                 best_params[1][3], limits, 100)
print('D=10 mean work time: {} seconds.'.format((time.time() - start) / kTimes))
```

```
D=10 mean work time: 6.585433387756348 seconds.
```

```
kTimes = 3
start = time.time()
for i in range(kTimes):
    bp_proc100, bp_log100, _ = full_process(best_params[2][0], best_params[2][1], best_params[2][2],
                                            best_params[2][3], limits, 300)
print('D=100 mean work time: {} seconds.'.format((time.time() - start) / kTimes))
```

```
D=100 mean work time: 875.7185890674591 seconds.
```

In [34]:
```
bp_log2[-1], bp_log10[-1], bp_log100[-1]
```

Out[34]:
```
(0.0, 0.0, 93.82838291139522)
```

In [42]:
```
print(opt_plan(bp_proc2))
```

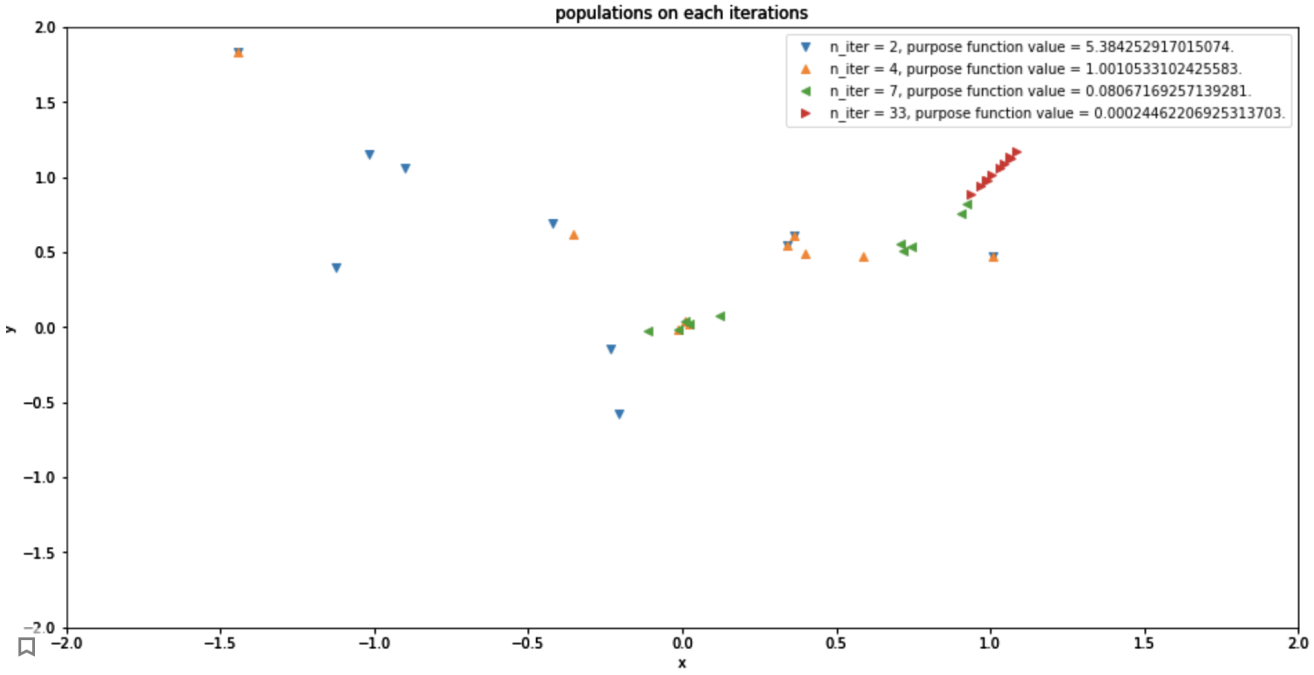
```
[1. 1.]
```

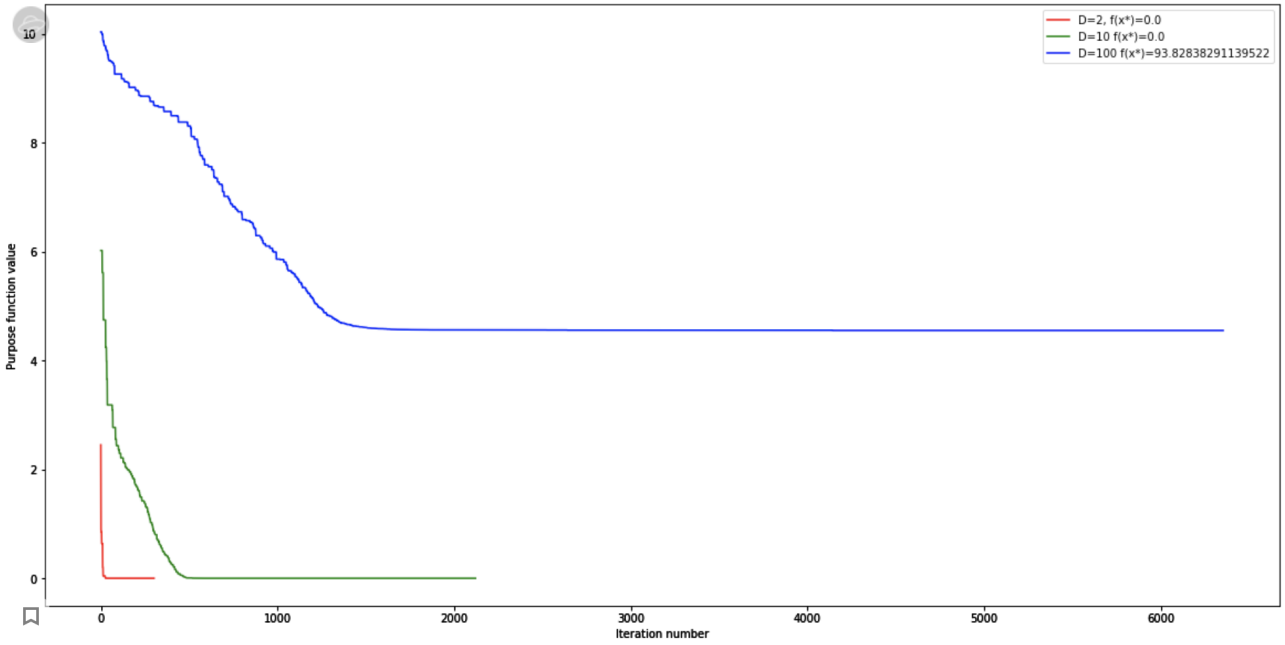
In [43]:
```
print(opt_plan(bp_proc10))
```

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

In [44]:
```
print(opt_plan(bp_proc100))
```

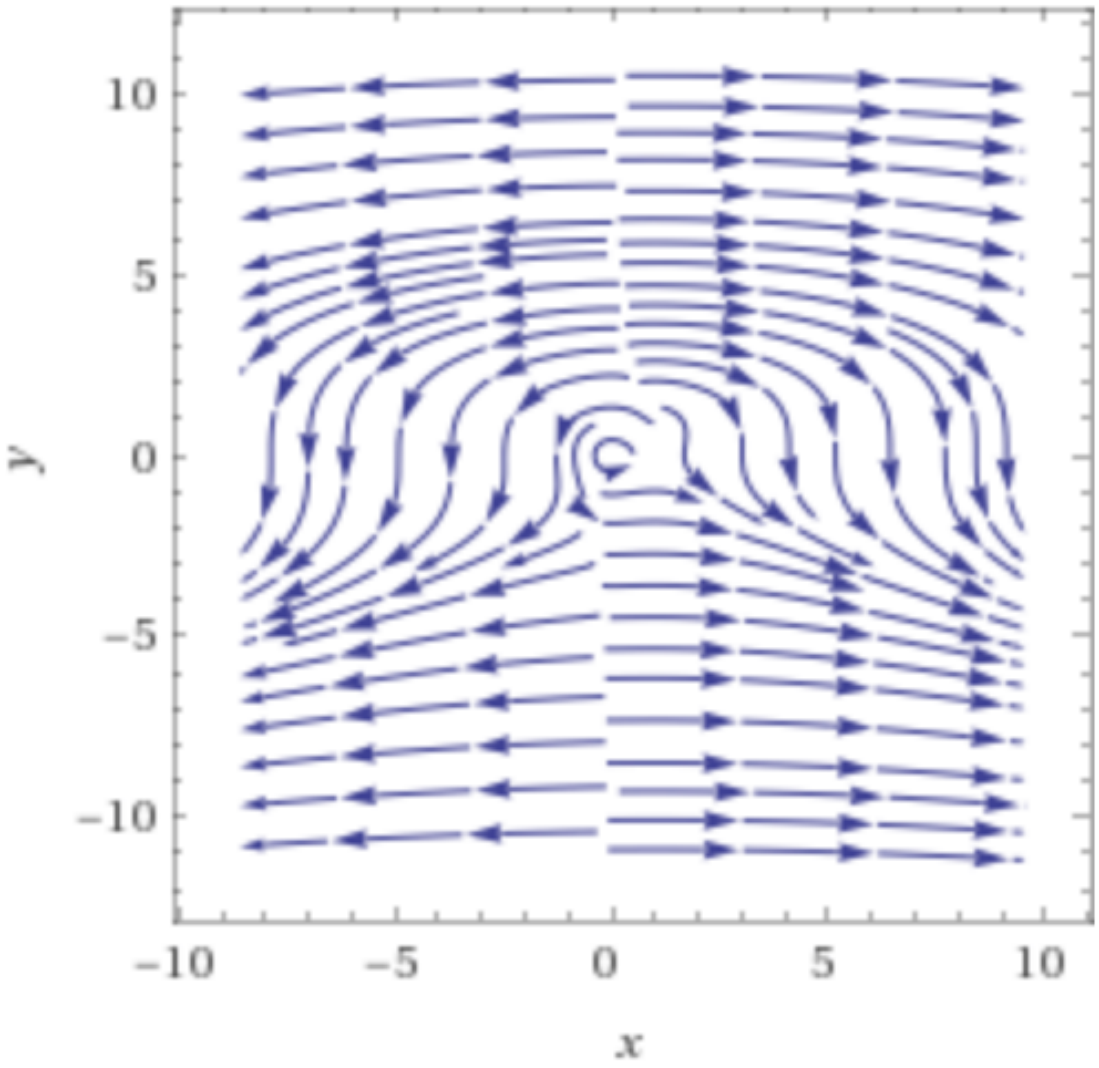
```
[6.52412292e-01 4.32555678e-01 1.95869173e-01 4.68087712e-02
 1.18971410e-02 1.02722695e-02 1.02102353e-02 1.02084331e-02
 1.02084364e-02 1.02084230e-02 1.02084199e-02 1.02084229e-02
 1.02084232e-02 1.02084292e-02 1.02084205e-02 1.02084182e-02
 1.02084412e-02 1.02084235e-02 1.02084199e-02 1.02084280e-02
 1.02084209e-02 1.02084242e-02 1.02084348e-02 1.02084183e-02
 1.02084179e-02 1.02084155e-02 1.02084244e-02 1.02084300e-02
 1.02084286e-02 1.02084280e-02 1.02084213e-02 1.02084301e-02
 1.02084288e-02 1.02084304e-02 1.02084207e-02 1.02084266e-02
 1.02084222e-02 1.02084239e-02 1.02084291e-02 1.02084258e-02
 1.02084282e-02 1.02084234e-02 1.02084327e-02 1.02084303e-02
 1.02084212e-02 1.02084232e-02 1.02084331e-02 1.02084175e-02
 1.02084201e-02 1.02084268e-02 1.02084173e-02 1.02084269e-02
 1.02084156e-02 1.02084260e-02 1.02084320e-02 1.02084235e-02
 1.02084231e-02 1.02084260e-02 1.02084202e-02 1.02084179e-02
 1.02084147e-02 1.02084222e-02 1.02084250e-02 1.02084297e-02
 1.02084259e-02 1.02084141e-02 1.02084207e-02 1.02084269e-02
 1.02084144e-02 1.02084256e-02 1.02084303e-02 1.02084239e-02
 1.02087332e-02 1.02635049e-02 2.62446420e-01 9.45578278e-01
 1.30399898e+00 1.27346240e+00 1.34715724e+00 1.11673598e+00
 5.14288335e-01 2.39351135e-01 2.02440380e-02 1.02101359e-02
 1.02084427e-02 1.02084200e-02 1.02084271e-02 1.02084153e-02
 1.02084204e-02 1.02084160e-02 1.02084210e-02 1.02084183e-02
 1.02084247e-02 1.02084310e-02 1.02084227e-02 1.02084134e-02
 1.02083528e-02 1.02042088e-02 1.00040815e-02 1.00082113e-04]
```


populations on each iterations
n_iter = 2, purpose function value = 5.384252917015074.
n_iter = 4, purpose function value = 1.0010533102425583.
n_iter = 7, purpose function value = 0.08067169257139281.
n_iter = 33, purpose function value = 0.00024462206925313703.
y
x


D=2, f(x*)=0.0
D=10 f(x*)=0.0
D=100 f(x*)=93.82838291139522
Purpose function value
Iteration number

| stream plot | $(-y + x\,y^2, x - x^2)$ |
|---|---|

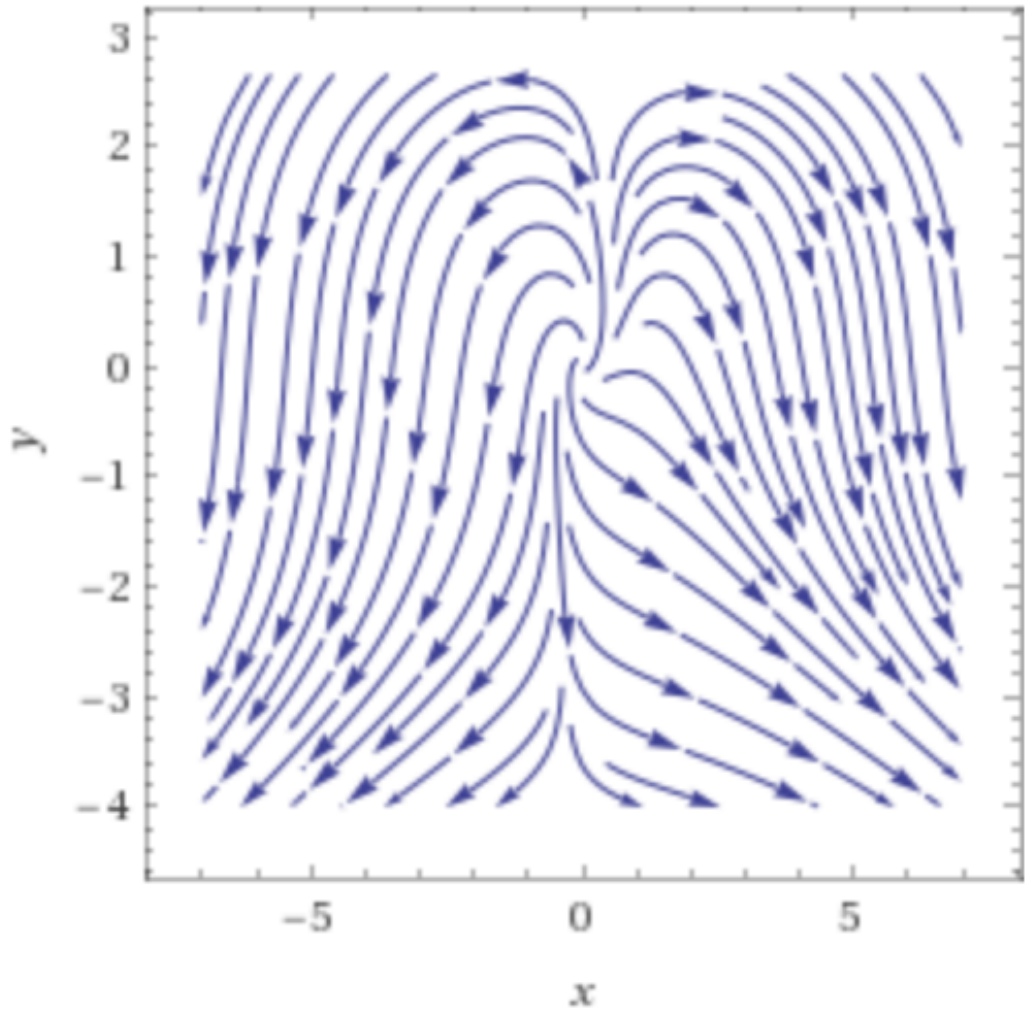
Plot:



Input interpretation:

| | |
|---|---|
| stream plot | $(x - y + x\,y^2,\ x - x^2 + y)$ |

Plot:



## Input interpretation:

| | |
|---|---|
| stream plot | $(-x - y + x y^2, x - x^2 - y)$ |

## Plot: