

Техническое задание на разработку процедуры решения нелинейного уравнения или системы нелинейных уравнений.

1. Название: «Процедура решения нелинейного уравнения или системы нелинейных уравнений»
2. Плановые сроки: 16.10.19
3. Назначение и область применения: Код процедуры будет использоваться как составная часть МПОП (модуль предварительной оценки параметров) для поиска варьируемых параметров радиолокатора, удовлетворяющих функциональному значению (значениям), заданным пользователем. Предполагается что код процедуры будет содержать формирование функционала качества для решаемой системы нелинейных уравнений (или одного нелинейного уравнения). Гибкость работы процедуры достигается тем, что число варьируемых и фиксированных переменных является переменным параметром задаваемым при вызове процедуры.
4. Общие требования к оформлению кода:
  - а) Код пишется на языке Python3 (версии 3.5+) с использованием стандарта форматирования текста PEP8;
  - б) Код выкладывается на общем GitHub репозитории;
  - в) написание юнит-тестов (TDD - test-driven development);
  - г) использование сторонних библиотек для задач минимизации сформированного функционала качества;
  - д) использование сторонних библиотек для поиска решений системы нелинейных уравнений.
5. Общие требования к документированию результатов: Научно-технический отчет в свободной форме с описанием основных методов решения системы нелинейных уравнений, применяемых подходов при решении системы и формирования функционалов качества.
6. Требования к процедуре:

Разрабатываемая процедура должна иметь следующие входные параметры:

  - а) `funcs ([func,])`- список ссылок на функции, задающие систему уравнений;

- b) `variable_parameters ([[string, ], ])`- список списков текстовых имен варьируемых параметров для каждого уравнения системы, если имя переменной одинаковое значит она общая для данных уравнений;
- c) `y_values ([float, ])`- список функциональных значений;
- d) `fixed_x_dict ([dict(), ])` - список словарей с именами и значениями фиксированных параметров для каждого уравнения системы.
- e) `bounds ([tuple, ])` - список списков с ограничениями на значения варьируемых параметров. Необязательный параметр.

Разрабатываемая процедура должна иметь следующие выходные параметры:

- a) `[dict(), ] | None` - функция возвращает список словарей с именами переменных и найденными решениями системы или None если решение не найдено.

7. Разрабатываемая процедура должна вызываться следующим образом:

`result_func(`

- a) `funcs = [f1, f2]` — список ссылок на функции, задающие систему уравнений;
- b) `variable_parameters = [[x, z], [x, w]]` — список списков текстовых имен варьируемых параметров для каждого уравнения системы, если имя переменной одинаковое значит она общая для данных уравнений;
- c) `y_values = [1, 2]` — числовой список (float) функциональных значений;
- d) `fixed_x_dict = [{t : 1, r : 2}, {t : 1, q : 3}]` — список словарей с текстовыми именами и значениями (float) фиксированных параметров для каждого уравнения системы

8. Разрабатываемая процедура должна возвращать словарь вида:

`{x:float, z:float}, {x:float, w:float}]`

9. Время работы разрабатываемой процедуры должно быть менее одной секунды.

10. Последовательность разработки должна быть следующей:

- a) Выбор и согласование библиотеки (функции) для решения системы, минимизации функционала;

- b) Выбор и согласование функций (набора функций для составления системы уравнений) решение которых необходимо будет найти;
- c) Реализация требуемого функционала.

## Приложение

В общем случае решается система уравнений вида:

$$F(X_{fixed}, X_{variable}) = Y$$

где:

1.  $F$  - система уравнений  $(f_1, f_2, \dots, f_n)$ ;
2.  $X_{fixed}$  - вектор фиксированных параметров;
3.  $X_{variable}$  - вектор варьируемых параметров;
4.  $Y$  - вектор функциональных значений.