

УДК 621.396

Анализ эффективности средств пакета *Scilab* при решении задач оптимизации¹

Семенова Т. И., Шакин В. Н., Фриск В. В., Кудряшова А. Ю.

Московский технический университет связи и информатики (МТУСИ)

ул. Авиамоторная, 8а, Москва, 111024, Российская Федерация

semtata@yandex.ru, victor.shakin@yandex.ru, frisk@mail.ru, asykka@bk.ru

Получено: 5 октября 2020 г.

Отрецензировано: 13 октября 2020 г.

Принято к публикации: 16 октября 2020 г.

Аннотация: В данной работе рассматриваются решатели *optim*, *fminsearch* и *plot*, включенные в библиотечные функции математического пакета *Scilab*, с точки зрения скорости и трудоемкости их применения при решении задач оптимизации на примерах функций Розенброка различных порядков. Приведен пример использования описанных решателей при расчете электрических цепей, найдено оптимальное значение локального минимума падающего напряжения вдоль длинной линии как функции одной переменной. Сделаны выводы об эффективности применения разных решателей математического пакета *Scilab* для расчетов цепей.

Ключевые слова: математические пакеты, *Scilab*, оптимизация, целевая функция, решатель, функции Розенброка, алгоритм Нелдера — Мида, метод *BFGS*, итерация, критерий эффективности.

Для цитирования (ГОСТ 7.0.5—2008): Семенова Т. И., Шакин В. Н., Фриск В. В., Кудряшова А. Ю. Анализ эффективности средств пакета *Scilab* при решении задач оптимизации // *Инфокоммуникационные и радиоэлектронные технологии*. 2020. Т. 3, № 2. С. 149—161.

Для цитирования (ГОСТ 7.0.11—2011): Семенова, Т. И. Анализ эффективности средств пакета *Scilab* при решении задач оптимизации / Т. И. Семенова, В. Н. Шакин, В. В. Фриск, А. Ю. Кудряшова // *Инфокоммуникационные и радиоэлектронные технологии*. — 2020. — Т. 3, № 2. — С. 149—161.

¹ Статья является расширенной версией доклада, представленного на 30-й Международной Крымской конференции «СВЧ-техника и телекоммуникационные технологии» — КрыМиКо'2020 (Севастополь, РФ, 6—12 сентября 2020 г.).

Analysis of the effectiveness of *Scilab* software tools in solving optimization problems

T. I. Semenova, V. N. Shakin, V. V. Frisk, and A. Y. Kudryashova
Moscow Technical University of Communications and Informatics
8a Aviamotornaya Str., Moscow, 111024, Russian Federation
semtata@yandex.ru, victor.shakin@yandex.ru, frisk@mail.ru, asykka@bk.ru

Received: October 5, 2020

Peer-reviewed: October 13, 2020

Accepted: October 16, 2020

Abstract: *This paper considers the optim, fminsearch and nmpplot solvers included in the library functions of the Scilab mathematical software from the point of view of the speed and complexity of their application in solving optimization problems using examples of Rosenbrock functions of various orders. An example of using the described solvers for calculating electrical circuits is given; the optimal value of the local minimum of the falling voltage along a long line as a function of one variable is found. Conclusions are made about the effectiveness of using different solvers of the Scilab mathematical software for circuit calculations.*

Keywords: *mathematical software, Scilab, optimization, objective function, solver, Rosenbrock functions, Nelder–Mead algorithm, BFGS method, iteration, efficiency criterion.*

For citation (IEEE): T. I. Semenova, V. N. Shakin, V. V. Frisk, and A. Y. Kudryashova, “Analysis of the effectiveness of Scilab software tools in solving optimization problems,” *Infocommunications and Radio Technologies*, 2020, vol. 3, no. 2, pp. 149–161. (In Russ.).

1. Введение

Любой исследуемый объект или процесс характеризуется своими конкретными параметрами (размерами, свойствами, показателями). В математическом моделировании взаимосвязь между параметрами описывается функциональными зависимостями. При проектировании рабочего узла конкретного объекта, или даже при организации и управлении производственным процессом, нас интересует наилучший вариант решения проблемы, то есть такой набор значений параметров целевой функции, описывающий моделируемый объект, который обеспечивает, как правило, ее экстремум.

Решение реальной задачи оптимизации далеко не всегда является тривиальной [1]. Это связано с тем, что количество параметров функции может быть достаточно велико, а кроме того на них могут быть наложены ограниче-

ния. Именно поэтому средства решения задач оптимизации включены в программное обеспечение практически всех современных математических пакетов [2, 3, 4]. В статье рассматриваются средства оптимизации математического пакета *Scilab*, современного, мощного, но в тоже время свободно распространяемого на сегодняшний день математического пакета. Они позволяют решать сложные и трудоемкие задачи оптимизации практически за несколько секунд, обеспечивающие при этом требуемую точность вычислений.

2. Основная часть

В пакете *Scilab* для решения задач оптимизации функций предназначен специальный набор средств, которые называются решателями [3]. Под решателем здесь понимается набор функций и алгоритмов, объединенных общим именем, предназначенный для решения конкретной задачи, в данном случае задачи оптимизации. Следует отметить, что некоторые функции, входящие в состав этого класса решателей, имеют и самостоятельное назначение (например, функция *costf*, предназначенная для вычисления производных).

В *Scilab* имеется несколько решателей, предназначенных для решения задач оптимизации. Каждый из них на выходе позволяет получить набор значений аргументов целевой функции, при которых она достигает своего оптимального значения. Кроме того, каждый решатель имеет и свой дополнительный (необязательный) набор параметров, по которым можно судить об успешном окончании процесса оптимизации, количестве итераций, количестве обращений к функции и многое другое. Исследование значений этих параметров и позволяет оценить эффективность применения решателя при решении конкретной задачи.

Исследования эффективности применения трех наиболее часто используемых решателей (*optim*, *fminsearch* и *nmplot*) были проведены на примере функций Розенброка от одной, двух и пяти переменных, которые являются тестовыми функциями для многих методов оптимизации:

$$f(x) = (x - 1)^2 \quad (1)$$

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (2)$$

$$f(x_1, x_2, x_3, x_4, x_5) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2 + 100(x_3 - x_4)^2 + (x_4 - 1)^2 \quad (3)$$

Отметим, что функция Розенброка относится к разряду овражных функций, для которых результат поиска сильно зависит от выбора началь-

ных условий, поэтому целевую функцию целесообразно предварительно исследовать (аналитически, таблично или графически). Так, например, для функции Розенброка от двух переменных $f(x_1, x_2)$ можно построить график линий уровней [5], где в качестве начального приближения можно взять точку, расположенную на линии уровня, отмеченную наименьшим значением функции, например, с координатами $(0,0)$, что представлено на рисунке 1.

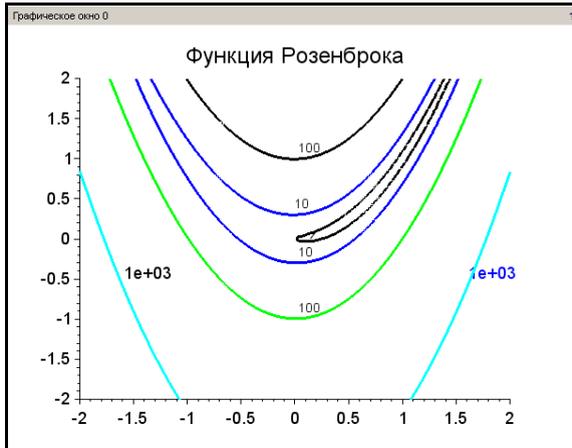


Рис. 1. Графическое представление линий уровня функции Розенброка от двух переменных.

Fig. 1. Graphical representation of the level lines of the Rosenbrock function in two variables

Optim — это решатель, предназначенный для решения задач как условной, так и безусловной нелинейной оптимизации. В основу алгоритмов этого решателя положены квази-ньютоновские методы. Эти методы представляют собой смешанную процедуру поиска с использованием квадратичного или кубического полиномов и алгоритм BFGS для аппроксимации и корректировки гессиана. Решатель *optim* использует внешнюю функцию *costf*, которая позволяет вычислять гессиан даже в тех случаях, когда вторые частные производные аналитически получить достаточно проблематично. Общий формат решателя *optim* имеет следующий вид:

$$[fopt, xopt [, gopt, work, iters, evals, err], ti, td] = optim(costf, x0, <contr>, algo, ...)$$

Обязательными входными параметрами решателя являются: *costf* — функция, которая вычисляет значение функции в точке x и находит направление градиента и x_0 — вектор начальных значений аргументов функции (начальная точка поиска). К необязательным входным параметрам относятся: *<contr>* — список параметров, содержащий границы изме-

нения переменных, *algo* — строка, фиксирующая используемые алгоритмы и т. д. [4]. Каждый алгоритм, реализованный в *optim*, имеет свои собственные критерии завершения, которые используют параметры, полученные от пользователя. При использовании входных параметров, представляющих собой списки, требуется предварительная настройка.

Очевидно, что представляющими интерес для пользователя выходными параметрами являются: *fopt* — значение целевой функции в точке оптимума, *hopt* — вектор аргументов, обеспечивших оптимальное значение функции и *gopt* — градиент целевой функции в точке *hopt*. Такие выходные параметры как *ti* — количество итераций, *td* — количество обращений к целевой функции и многие другие [4] используются в основном более узкими специалистами.

В основу решателя *fminsearch* положен алгоритм Нелдера — Мида, основанный на обновлении симплекса, представляющего собой набор из $k \geq n + 1$ вершин (n — количество переменных), где каждая вершина связана с одной точкой и одним значением функции. Общий вид формата решателя *fminsearch* [3]:

$$[optx, optf, exitflag, output] = fminsearch(costf, x0, options)$$

Входными параметрами решателя являются: *costf* — функция, которая вычисляет значение функции в точке x и находит направление градиента; x_0 — вектор начальных значений аргументов функции (начальная точка поиска); *options* — является списком необязательных настроек. В этот список входят такие параметры как: *options.MaxIter* — максимальное количество итераций, *options.MaxFunEvals* — максимальное количество оценок целевой функции, *options.TolX* — абсолютный допуск на симплекс размер и многие другие важные настройки.

Возвращаемые значения параметров: *optx* — вектор значений переменных, при которых достигается оптимум функции; *fval* — значение функции для набора найденных значений аргументов; *exitflag* — флаг причины остановки оптимизации; *output* — структура, хранящая подробную информацию о выходе из алгоритма, где в качестве причины остановки вычислений выводится: -1 — достигнуто максимальное количество итераций, 0 — достигнуто ограничение по количеству итераций или 1 — найдено решение в пределах погрешности.

Кроме описанных выше решателей *optim* и *fminsearch*, в *Scilab* интерес представляет и решатель *nmpplot* [3], который использует алгоритмы

оптимизации прямого поиска и включает несколько методов оптимизации прямого поиска, основанных на симплексном методе. При этом всю информацию о процессе оптимизации на каждой итерации можно отобразить на экране или сохранить в файлах для их последующего анализа. Этими данными могут быть: значения координат симплекса; значения функции, усредненные по вершинам; минимальные значения функции в симплексе; размеры симплекса.

На рисунке 2 приведен пример графической траектории поиска минимума функции Розенброка от двух переменных с использованием решателя *fminsearch*. Здесь выбор направления спуска определяется симплексом. Отметим, что самые большие изменения в значении функции происходят на первых итерациях, что характерно для всех градиентных методов. Подробно пример использования решателя *fminsearch* рассмотрен в [3].

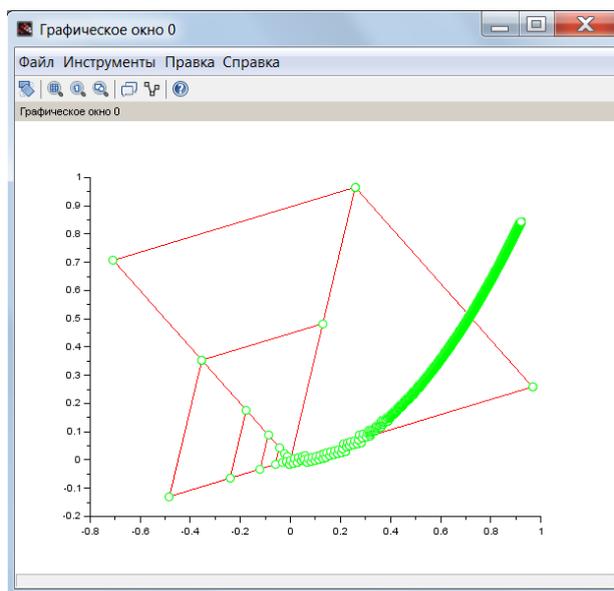


Рис. 2. Траектория спуска поиска минимума функции от двух переменных с использованием решателя *fminsearch*.

Fig. 2. Descent trajectory of finding the minimum of a function of two variables using the *fminsearch* solver

В качестве простейшего примера рассмотрим простую задачу одномерного поиска, часто встречающуюся в электрических цепях: найти угловую частоту ω , при которой полное сопротивление последовательного колебательного *RLC*-контура минимально, схема представлена на рисунке 3.

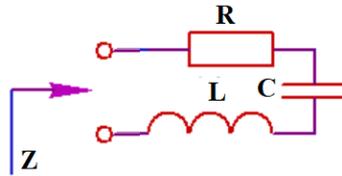


Рис. 3. Схема последовательного колебательного RLC -контура.

Fig. 3. Diagram of a series-oscillating RLC circuit

Функция полного сопротивления $Z(\omega)$, как известно из теории электрических цепей [6, 7], будет следующей:

$$Z(\omega) = \sqrt{R^2 + \left(\omega L - \frac{1}{\omega C}\right)^2} \quad (4)$$

Здесь $Z(\omega)$ — функция полного сопротивления RLC -контура; ω — угловая частота; R — сопротивление контура; L — индуктивность контура; C — емкость контура.

Ниже рисунке 4 приведен программный код для построения графика функции полного сопротивления $Z(\omega)$ от угловой частоты $\omega = [0,05, 0,5]$ рад/с при $L = 10$ Гн, $C = 10$ Ф и $R = 0,5$ Ом, а также вычисление оптимального значения частоты ω_{\min} и значения комплексного сопротивления двумя методами оптимизации (*optim* и *fminsearch*) при $\omega_0 = 0,005$. Сам график представлен на рисунке 5.

С использованием описанных выше решателей также были получены оптимальные значения функций Розенброка от одной, двух и пяти переменных. За начальные значения параметров для всех функций были приняты равные нулю значения. Время выполнения поиска минимума функций получены с использованием команд включения и выключения таймера (*tic* и *toc*), добавленные в вычисление соответственно до и после обращения к решателям. Для получения значений количества итераций и количества обращений к целевой функции для каждого решателя был использован расширенный параметр, позволяющий получить эти показатели в качестве выходных параметров. За точность вычислений оптимума принят порог по градиентной норме, по умолчанию равный $\%eps = 2.2 \text{ D-}16$. Информация представлена в таблице 1.

По результатам расчета можно сделать вывод, что по всем значениям показателей, приведенных в таблице 1, решатели *fminsearch* и *nplot* проигрывают решателю *optim*. Однако, несмотря на это, решатели *fminsearch* и *nplot* имеют множество таких полезных свойств как: вывод

```

1 // График полного сопротивления последовательного RLC-контура
2 // Исходные данные
3 L = 10; // Индуктивность, Гн
4 C = 10; // Емкость, Фр
5 R = 0.5; // Сопротивление, Ом
6 // Диапазон изменения угловой частоты, рад/с
7 w=[0.05:0.001:0.5];
8 Z = sqrt(R^2+(w*L-1./(w*C)).^2); // Полное сопротивление, Ом
9
10 plot(w,Z) // Построение графика
11 // Подписи графика и осей
12 xtitle("Полное сопротивление последовательного RLC-контура", "w, рад/с", "Z(w), Ом")
13 xgrid // Координатная сетка
14 // Поиск минимума угловой частоты, при которой наблюдается
15 // минимум полного сопротивления последовательного RLC-контура
1 function Z = costf(w)
2     funcprot(0)
3     // Исходные данные
4     L = 10; // Индуктивность, Гн
5     C = 10; // Емкость, Фр
6     R = 0.5; // Сопротивление, Ом
7     Z = sqrt(R^2+(w*L-1./(w*C)).^2); // Функция полного сопротивления
8 endfunction
24 // Описание целевой функции
1 function y = Z(w)
2     funcprot(0)
3     // Исходные данные
4     L = 10; // Индуктивность, Гн
5     C = 10; // Емкость, Фр
6     R = 0.5; // Сопротивление, Ом
7     y = sqrt(R^2+(w*L-1./(w*C)).^2); // Функция полного сопротивления
8 endfunction
33 // Функция, вычисляющая целевую функцию и ее производную в заданной точке
1 function [ff,g,int] = costf(w, int)
2     ff = Z(w)
3     g = numderivative(Z, w)
4 endfunction
38 [Zmin, Wmin] = optim(costf, w0); // Обращение к функции optim
39 mprintf('Результат функции optim: Zmin=%g·Wmin=.%g', Zmin, Wmin);
40 [Wmin, Zmin,output] = fminsearch(costf,w0); // Обращение к функции fminsearch
41 mprintf('Результат функции fminsearch: Zmin=%g·Wmin=.%g\n', Zmin, Wmin)
42 //endfunction
43

```

Рис. 4. Вычисление оптимального значения частоты и значения комплексного сопротивления решателями *optim* и *fminsearch*.

Fig. 4. Calculation of the optimal frequency and complex impedance using the *optim* and *fminsearch* solvers

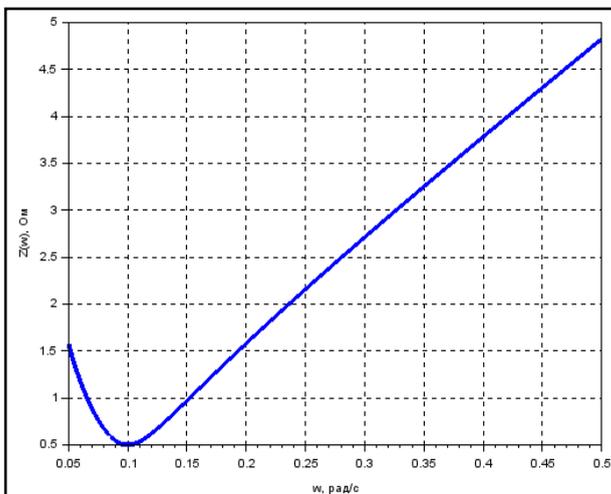


Рис. 5. График функции полного сопротивления $Z(\omega)$ от угловой частоты ω .

Fig. 5. Graph of the impedance function $Z(\omega)$ versus the angular frequency ω

Табл. 1. Оптимальные значения функций Розенберка

Решатель	Количество переменных в функции	Количество итераций	Количество обращений к целевой функции	Время (сек.)
optim	1	2	4	0.043
	2	20	27	0.084
	5	34	49	0.138
fminsearch	1	20	40	0.894
	2	67	124	2.675
	5	302	492	11.578
nmpplot	1	39	78	8.127
	2	358	642	11.256
	5	1332	2674	50.117

графиков и возможность получения на каждой итерации подробной информации о множестве числовых показателей, отражающих состояние процесса оптимизации. При этом стоит отметить, что решатель *fminsearch* отличается простотой настройки параметров.

Таким образом, в случаях, если перед пользователем пакета *Scilab* стоит задача нахождения только координат точки экстремума функции, целесообразнее использовать решатель *optim*, поскольку очевидно, что он эффективнее с точки зрения времени выполнения поиска, а его базовые настройки вполне подходят под большинство функций [5]. Если же необходимо иметь информацию о промежуточных результатах, получить графическую интерпретацию процесса оптимизации, то есть получить полную картину, позволяющую проанализировать данные в процессе оптимизации функции, целесообразнее использовать решатели *fminsearch* или *nmpplot*.

В качестве примера оптимизационной задачи рассмотрим поиск локального минимума величины напряжения как функции одной переменной.

Проведем анализ распределения падающей волны напряжения вдоль длинной линии, схема показана на рисунке 6.

Функция напряжения $u(x)$ от координаты x выражается формулой:

$$u(x) = U_m e^{-\alpha x} \sin(\omega t - \beta x + \psi) \tag{5}$$

где $U_m = 1$ В — амплитуда;

$\alpha = 0,11/\text{м}$ — коэффициент ослабления;

$\beta = 0,1$ рад/м — коэффициент фазы;
 $l = 5$ м — длина линии;
 $\omega = 1$ рад/с — угловая частота;
 $t = 5$ с — фиксированный момент времени;
 $\psi = 0$ рад — фаза.

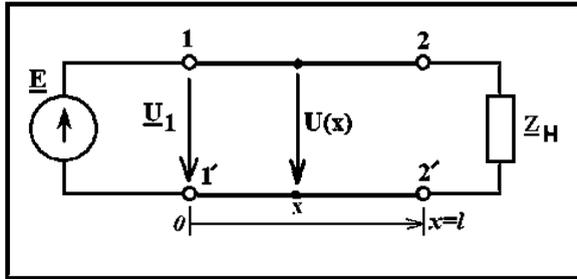


Рис. 6. Схема длинной линии.

Fig. 6. Long line diagram

На рисунке 7 приведена программа для построения графика падающего напряжения вдоль длинной линии $u(x)$ от координаты x . На рисунке 8 представлен график распределения мгновенного напряжения падающей волны вдоль длинной линии, из которого видно, что функция мощности имеет минимум в районе точки 2,4 м.

```

--> // Incident wave
--> // Initial data
--> a=0.1; // Coefficient of attenuation, Ohm
--> b=2.6; // Phase ratio, rad/m
--> Um=1; // Amplitude, V
--> l=5; // Line length m
--> w=1; // Angular frequency, rad/s
--> t=5; // Fixed time, s
--> // Coordinate x, m
--> x=[2:1/100:1];
--> // Incident wave, V
--> u=Um*exp(-a*x).*sin(w*t-b*x);
--> // Schedule
--> plot(x,u)
--> // Graph and axis captions
--> xtitle("Incident wave along a long line","x, m","u(x), V")
--> xgrid // Coordinate grid
  
```

Рис. 7. Программа для построения графика падающего напряжения вдоль длинной линии $u(x)$ от координаты x .Fig. 7. Program for plotting the voltage drop along the long line $u(x)$ from the x coordinate

Для нахождения более точного значения координаты x , при которой наблюдается минимум напряжения, воспользуемся, описанными выше решателями *optim* и *fminsearch*. Программа для уточнения оптимальной координаты x_0 и полученные результаты ее выполнения приведена на рисунке 9.

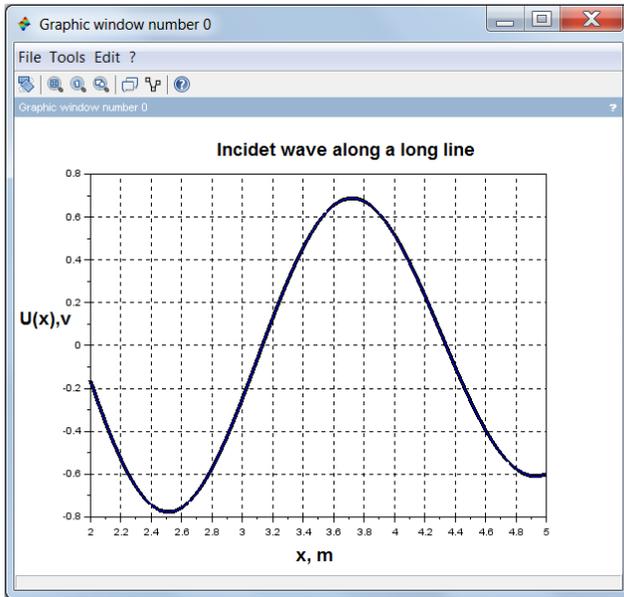


Рис. 8. График падающего напряжения вдоль длинной линии $u(x)$ от координаты x .

Fig. 8. Graph of the voltage drop along the long line $u(x)$ from the x coordinate

```
--> // Search for the coordinates at which
--> // minimum voltage of the incident wave
--> function U=costf(x)
  > funcprot(0)
  > // Initial data
  > a=0.1; // Coefficient of attenuation, Ohm
  > b=2.6; // Phase ratio, rad/m
  > Um=1; // Amplitude, V
  > l=5; // Line length m
  > w=1; // Angular frequency, rad/s
  > t=5; // Fixed time, s
  > U=Um*exp(-a*x).*sin(w*t-b*x); // Incident wave, V
  > endfunction
-->
--> x0=[2.4]; // nitial approximation x0, m
--> //The coordinate at which u(x0)->min
--> [x0 Umin]=fminsearch(costf,x0);
--> mprintf('Result of the fminsearch function:x0=%g Umin=%g',x0,Umin)
-->
--> //The coordinate at which u (x0) -> min
--> [x0 Umin]=optim(costf,x0);
--> mprintf('Result of the optim function:x0=%g Umin=%g',x0,Umin)

The result of the fminsearch function:
  x0 = 2.5124414  Umin = -0.7772576

The result of the function optim:
  x0 = 2.5124414  Umin = -0.7772576
```

Рис. 9. Программа для уточнения оптимальной координаты x_0 и полученные результаты ее выполнения.

Fig. 9. Program for specifying the optimal coordinate x_0 and the results of its implementation

Оба решателя дали один и тот же результат, а расчет показал, что минимум падающего напряжения будет равен $U_{\min} = -0.7772576$ В при $x_0 = 2.5124414$ м, что полностью согласуется с теорией электрических цепей [7, 8].

3. Заключение

Таким образом, на примере решения задачи из области электротехники удалось установить, что программный продукт *Scilab* предоставляет все функциональные возможности, необходимые пользователю для решения таких задач.

При решении с использованием ручных расчетов время, затрачиваемое на поиск решения проблемы, увеличится в десять раз. При проектировании рабочего блока того или иного объекта или даже при организации и управлении производственным процессом такое увеличение времени на поиск решения весьма критично. Современное программное обеспечение *Scilab* позволяет решать сложные и трудоемкие задачи оптимизации за несколько секунд, обеспечивая при этом требуемую точность вычислений. Таким образом, этот программный продукт может быть использован для решения сложных задач оптимизации в производстве.

В дальнейших исследованиях можно сравнить скорость и удобство решения задач оптимизации в различных программных системах, таких как *Matlab*, *Silab* и *Octav*.

Список литературы

1. Семенова Т. И., Кравченко О. М., Шакин В. Н. Вычислительные модели и алгоритмы решения задач численными методами : учебное пособие. М. : МТУСИ, 2017. 82 с.
2. Семенова Т. И., Загвоздкина А. В., Загвоздкин В. А. Изучение численных методов с использованием средств пакета Scilab // Экономика и качество систем связи. 2017. № 4 (6). С. 60—69.
3. Документация Scilab [Электронный ресурс]. Режим доступа: https://help.scilab.org/docs/6.0.1/ru_RU/index.html <http://jre.cplire.ru/jre/dec16/5/text.pdf> (дата обращения: 14.01.2020).
4. Шакин В. Н., Семенова Т. И., Фриск В. В. Базовые средства математического пакета Scilab. Учебник для вузов. М. : Горячая линия — Телеком, 2019. 336 с.
5. Семенова Т. И., Загвоздкина А. В., Загвоздкин В. А. Визуализация результатов вычислений в Scilab // Информатика и кибернетика. 2018. № 4 (14). С. 5—11.
6. Фриск В. В. Основы теории цепей. М. : РадиоСофт, 2002. 288 с.
7. Смирнов Н. И., Фриск В. В. Теория электрических цепей. Учебник для вузов. М. : Горячая линия — Телеком, 2019. 286 с.
8. Adzhemov A. S., Kudryashova A. Y. Evaluation program of an efficient source coding algorithm under the condition of converting metric spaces // В сб. : 2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF 2019). 2019. С. 8840609.

Информация об авторах

Кудряшова Анастасия Юрьевна, аспирант, ассистент кафедры общей теории связи Московского технического университета связи и информатики (МТУСИ), г. Москва, Российская Федерация. ORCID 0000-0002-3673-7667.

Семенова Татьяна Игоревна, к. т. н., доцент кафедры информатики Московского технического университета связи и информатики (МТУСИ), г. Москва, Российская Федерация.

Шакин Виктор Николаевич, к. т. н., доцент кафедры информатики Московского технического университета связи и информатики (МТУСИ), г. Москва, Российская Федерация.

Фриск Валерий Владимирович, к. т. н., доцент кафедры теории электрических цепей Московского технического университета связи и информатики (МТУСИ), г. Москва, Российская Федерация.

Information about the authors

Anastasiya Y. Kudryashova, graduate student, assistant at the Department of General Theory of Communications, Moscow Technical University of Communications and Informatics (MTUCI), Moscow, Russian Federation. ORCID 0000-0002-3673-7667.

Tatyana I. Semyonova, Candidate of Technical Sciences, Associate Professor of the Department of Informatics of the Moscow Technical University of Communications and Informatics (MTUCI), Moscow, Russian Federation.

Viktor N. Shakin, Candidate of Technical Sciences, Associate Professor of the Department of Informatics of the Moscow Technical University of Communications and Informatics (MTUCI), Moscow, Russian Federation.

Valery V. Frisk, Candidate of Technical Sciences, Associate Professor of the Department of Theory of Electrical Circuits, Moscow Technical University of Communications and Informatics (MTUCI), Moscow, Russian Federation.