# Electric DC circuit as mathematical programming models

## Schemes, algorithms and programs

Solomon I. Khmelnik

In this book we are dealing with circuits DC, supplemented by Dennis Transformers, invertible elements, etc... It is shown that such circuits can be used as physical models of certain problems: linear and non-linear equations systems solution, convex programming problems. We are discussing fast methods and algorithms for calculating such electrical circuits and hence solve these problems. Programs for solving such problems are provided in MATLAB system.

ISBN 978-1-312-86948-6
90000
9 781312 869486
ID: 16261663
www.lulu.com

Electric DC circuit as mathematical programming models. Schemes, algorithms and programs.

Solomon I. Khmelnik

# Electric DC circuit as mathematical programming models

*Schemes, algorithms and programs*



Published by "MiC" - Mathematics in Computer Comp.
BOX 15302, Bene-Ayish, Israel, 60860
Fax: ++972-3-9471301
Printed in United States of America, Lulu Inc., ID 16261663
ISBN 978-1-312-86948-6

ISBN 978-1-312-86948-6
90000
9 781312 869486

Israel 2015

# Annotation

In this book we are dealing with circuits DC, supplemented by Dennis Transformers, invertible elements, etc... It is shown that such circuits can be used as physical models of certain problems: linear and non-linear equations systems solution, convex programming problems. We are discussing fast methods and algorithms for calculating such electrical circuits and hence solve these problems. Programs for solving such problems are provided in MATLAB system.

# Contents

# Preface

Dennis has proposed a theory of direct current electric circuits including resistors, diodes, current sources, voltage sources and the DC transformer or transformers instantaneous values [1]. Such transformers were explored first by Dennis as so we shall call them the *Dennis transformers* and denote them as **TD**. Dennis had showed that such electric circuits are simulating the quadratic programming problem with inequality constraints. Dennis proposed TD as an abstract mathematical model (for mathematical theory interpretation). However no methods of physical realization of TD have been presented. Due to the technical complexity of such implementation the circuits with DC transformers have not been used till today.

For the first time such a property of electrical circuits was noticed by Maxwell [2], who found that in circuits with resistors the currents are minimizing the power of heat loss. Then (as indicated above) Dennis had proved this property for sufficiently complex linear DC circuits.

The book contains **two** chapters and programs. **Chapter 1** discusses the linear DC circuit, comprising current and voltage sources, resistors and Dennis transformers. These circuits are models of quadratic programming problem with linear constraints. **In Chapter 2**, the electrical circuits are supplemented by diodes. It is shown that such DC circuits simulate convex programming problem with nonlinear constraints

In both chapters we deal with various mathematical problems that are modeled by the described DC circuits, and give the related programs. These programs can perform the following calculations:

1. Calculation of direct current electric circuits with resistors, transformers of instantaneous values, diodes, voltage and current sources.
2. Solution of the system of linear equations and inequalities.
3. Problem of solving quadratic and linear programming with constraints in the form of equalities and inequalities.
4. The solution of underdetermined and overdetermined system of linear equations.
5. The solution of some other problems.

This book is a translation of a part of the book [3].

# Chapter 1. Linear DC Circuits

## 1.1. Electric circuits with resistors

### 1.1.1. A Simple Electric Circuit.

Let us consider an *electric circuit with current source* and select in it three types of branches:

1. a branch with current source connected to the node,
2. a series circuit with resistors and voltage sources contained between two nodes,
3. a branch with a conductivity connected between a node and a common bus.

The current directed to the node will be considered positive. The number of branches of the type 2 will be denoted as *m*, the number of branches of type 1 – as *n*. Such electrical circuit is described by the following equations system:

$$N^T \cdot \varphi + R \cdot I - U = 0, \tag{1}$$

$$N \cdot I + C - i = 0, \tag{2}$$

where

$C$ - vectors of currents in the type 1 branches (dimension *n*);

$I$ - vectors of currents in the type 2 branches (dimension *m*);

$U$ – vector of the second type branches (dimension *m*);

$G$ – diagonal matrix of conductivities in the branches of third type (dimension *n* * *n*);

$i$ – currents vectors in the branches of third type;

$\varphi$ - vector of nodes potentials (dimension *n*);

$N$ – incidence matrix with the elements 1, 0, -1 (dimension *n** *m*);

$R$ - diagonal matrix of conductivities in the second type branches (dimension *m***m*);

$\phi$ - vector of voltages on the resistors in the second type branches (dimension *m*), and

$$\phi = R \cdot I. \tag{3}$$

In this system equation (2) describes the second law of Kirchhoff and equation (1) – the first law of Kirchhoff. In this system the known vectors are $C$ и *E,* and the sought vectors - *I* and $\varphi$.

Let us consider the function

$$F(I,i)=\frac{1}{2}\cdot I^T\cdot R\cdot I-U^T\cdot I+i\cdot\frac{1}{2G}\cdot i. \tag{4}$$

Let us find the necessary conditions of this function optimum with constraints of the form (2). They are of the form of equations (1), where $\varphi$ is a vector of undetermined Lagrange multipliers for the constraints (2), that appear when the optimized function is supplemented by a term $\varphi^T\cdot(N\cdot I+C-i)$

Further we have

$$\frac{\partial^2 F}{\partial\ I^2}=R,\ \frac{\partial^2 F}{\partial\ i^2}=\frac{1}{G}. \tag{5}$$

From this follows that the function (4) has a global minimum. Thus, the minimization of function (4) under the constraints in the form of the first law of Kirchhoff (2) leads to the equations second law of Kirchhoff (1). Consequently, the computation of a DC electric circuit is equivalent to finding a conditional maximum of the function (4).

Let us multiply (2) from the left by $\varphi^T$ and (1) – from the left by $I^T$. Then we shall find that from Kirchhoff laws follows the fact that the summary power of the electric circuit is equal to zero:

$$\varphi^T\cdot C-I^T\cdot R\cdot I+I^T\cdot U-\varphi^T\cdot i=0, \tag{6}$$

Now let us consider the function

$$\Phi(\phi,\varphi)=-\phi^T\frac{1}{2R}\phi+C^T\cdot\varphi-\frac{1}{2}\varphi^T\cdot G\cdot\varphi. \tag{7}$$

Apparently,

$$i=G\cdot\varphi \tag{8}$$

and the Kirchhoff laws may be rewritten in the form:

$$N^T\cdot\varphi+\phi-U=0, \tag{9}$$

$$N\cdot I+C-G\cdot\varphi=0, \tag{10}$$

Let us find the necessary conditions of this function's optimum under the constraints that are the equations of second Kirchhoff law in the form (9). These constraints have the form of equations (10) and (3), where *I* is the vector of undetermined Lagrange multipliers for the constraint (8), which appear when the optimized function is supplemented by the term $I^T \cdot \left(N^T \cdot \varphi + \phi - U\right)$.

Further we have

$$\frac{\partial^2 \Phi}{\partial\ \phi^2} = \frac{-1}{R}, \quad \frac{\partial^2 \Phi}{\partial\ \varphi^2} = -G.$$

From this it follows that the function (7) has a global optimum. Thus, maximization of the function (7) under constraints in the form of second Kirchhoff law equations (8) leads to the equations of the first Kirchhoff law (10) and the condition (3). Consequently, the computation of a DC electric circuit is equivalent to finding a conditional maximum of the function (7).

The problems of minimization of function (4) with constraint (2) and of maximization of function (7) with constraint (8) are dual problems

The equations of simple DC always have a unique solution.

### 1.1.2. Unconditional electric circuit.

Let us consider an electric DC circuit, which has $G \equiv 1/\rho$. We shall call such circuit unconditional (the meaning of such name will be clear from further discussion). Apparently, the resistors $\rho$ can be considered as additional branches of the electric circuit. We however shall use another method, allowing to describe such circuits in a more compact way and, as a result, to reduce substantially the dimension of vectors and matrices.

The vector $I$ of currents flowing through the resistors $\rho$, is connected with other currents of the circuits by correlation (2). Besides,

$$\varphi = \rho \cdot i \ . \tag{22}$$

Let us consider minimization problem of the function (4) with constraint (2). In this case (4) takes the following form:

$$F(I) = \frac{1}{2} \cdot I^T \cdot R \cdot I + \frac{\rho}{2} \cdot i^T \cdot i \ - U^T \cdot I \,. \tag{23}$$

Substituting (2) in (23), we get

$$F(I) = \frac{1}{2} \cdot I^T \cdot R_N \cdot I - U_N^T \cdot I \,, \tag{24}$$

where

$$R_N = R + \rho \cdot N^T \cdot N \,, \tag{25}$$

$$U_N = U - \rho \cdot N^T \cdot C \,. \tag{26}$$

Let us find the necessary conditions of the unconditioned minimum of function (24). They have the form of the following equations:

$$R_N \cdot I - U_N = 0 \tag{27}$$

or

$$(R + \rho \cdot N^T \cdot N) \cdot I - U + \rho \cdot N^T \cdot C = 0 \tag{28}$$

or

$$R \cdot I - U + \rho \cdot N^T \cdot (N \cdot I + C) = 0 \tag{29}$$

Taking into account (2) and (22), we get the equation of the second Kirchhoff law (1). Thereby, the unconditional minimization of function (24) formally gives the same result as the conditional minimization of function (4).

If $\rho \to \infty$, then $i \to 0$. This follows from the fact than in an electric circuit C the heat loss capacity is being minimized – see the term $\rho \cdot i^T \cdot i$ in (23). Thus, for $\rho \to \infty$ the result of unconditional minimization of function (24) from *I* approach the result of conditional function (4) of *I* and $\varphi$ minimization (the condition is the first Kirchhoff law). So,

1. Calculation of the electric DC circuit is equivalent to the calculation of the corresponding unconditional circuit at $\rho \to \infty$.
2. Calculation of an unconditional circuit is equivalent to unconditional optimization of the function (24), when the vector of currents *I* in the branches.
3. The problem of unconditional minimization has dimension *m*, while the problem of conditional optimization has the dimension *(m+n)*.
4. The nodes potentials are determined by the formula

$$\varphi = \rho \cdot (N \cdot I + C). \tag{30}$$

Mind that the first Kirchhoff (1.1.2) law is satisfied with a certain error.

### 1.1.3. The algorithm of calculation an unconditional electric circuit

The existence of global minimum permits us to use the gradient descent method for the electric circuit calculation. We shall outline the idea of this method for an unconditioned electric circuit. Идею метода рассмотрим для безусловной электрической цепи. It consists in the following. For given values of vector *I* its new value is calculated by the formula

$$I_n = I - a \cdot p, \tag{31}$$

where

*p*– the gradient of vector *I*,

*a* – the step along the gradient.

The gradient is:

$$p = R_N \cdot I - U_N \tag{32}$$

When changing the vector from *I* to $I_n$ the function (24) changes by the value $\Delta F = F(I_n) - F(I)$. Further we have:

$$\frac{\partial\ \Delta F}{\partial\ a} = \frac{\partial\ F(I_n)}{\partial\ a} = \frac{\partial\ F(I_n)}{\partial\ I_n} \cdot \frac{\partial\ I_n}{\partial\ a} = p^T \frac{\partial\ F(I_n)}{\partial\ I_n}.$$

The optimal value of the step is determined from the condition $\frac{\partial\ \Delta F}{\partial\ a} = 0$ or $p^T \frac{\partial\ F(I_n)}{\partial\ I_n} = 0$. Thus,

$$p^T \cdot (R_N \cdot (I - a \cdot p) - U_N) = 0.$$

From this we find

$$a = \frac{p^T \cdot R_N \cdot I - p^T \cdot U_N}{p^T \cdot R_N \cdot p} \tag{33}$$

After the transformation (33) taking into account (32) we find:

$$a = \frac{p^T \cdot p}{p^T \cdot R_N \cdot p}. \tag{34}$$

In this way the iteration process of finding the minimum of function (24) permits to find vector *I*. On each iteration:

- the gradient *p* is calculated by formula (32) for the given vector *I*;
- the coefficient *a* is calculated by (33) for given *p*;
- the new value of vector *I* is calculated by formula (31).

Iterative process continues till the value

$$\varepsilon_2 = p^T \cdot p \tag{35}$$

reaches a given minimum. Virtually we should strive to the value

$$\varepsilon_2 = \varepsilon \cdot U_N^T \cdot U_N, \tag{36}$$

where $\varepsilon << 1$ is a given value of relative error.

# 1.2. Electric Circuits with Dennis Transformers

## 1.2.1. A Simple Electric Circuit with Dennis Transformers

The electric circuits described below contain DC transformers or transformers of instantaneous values. The first who described them was Dennis [4]. So below they are called *Dennis transformers* and denoted as TD. Dennis proposed TD as an abstract mathematical construction (for interpreting a mathematical theory) and developed the theory of electric DC circuits with constant voltage, including TD, resistors, diodes, current and voltage sources. However no methods of physical realization of TD have been presented. Due to the technical complexity of such implementation the circuits with DC transformers have not been used till today.

TD has primary and secondary windings. Instantaneous values of currents and voltages in these windings are interconnected in the same way as the effective values of sinusoidal currents and voltages in a conventional transformer.

On Fig 1 TD is portrayed schematically. It comprises two branches - a primary branch with current $i_1$ and voltage $e_1$ and a secondary branch with current $i_2$ and voltage $e_2$. TD described by the equations:

$$i_1 + t \cdot i_2 = 0, \tag{0a}$$

$$e_2 - t \cdot e_1 = 0, \tag{0в}$$

where $t$ – transformation coefficient.

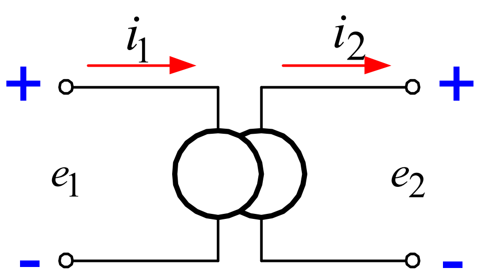



*Fig. 1. Schematic picture of TD.*

Out of these equations, it follows that $e_1 i_1 = -e_2 i_2$, i.e. the power supplied by the primary and secondary branches of the TD in the electrical circuit, in the sum give zero - so TD is a passive element. Dennis transformer TD can be seen as a node, where the currents are summed with weighting coefficients. Thus there is a complete analogy with the first Kirchhoff's law for the nodes.

In [3] various versions of physical realization of DT are presented. Thus *the electric circuits with DT* become *physically realizable.*

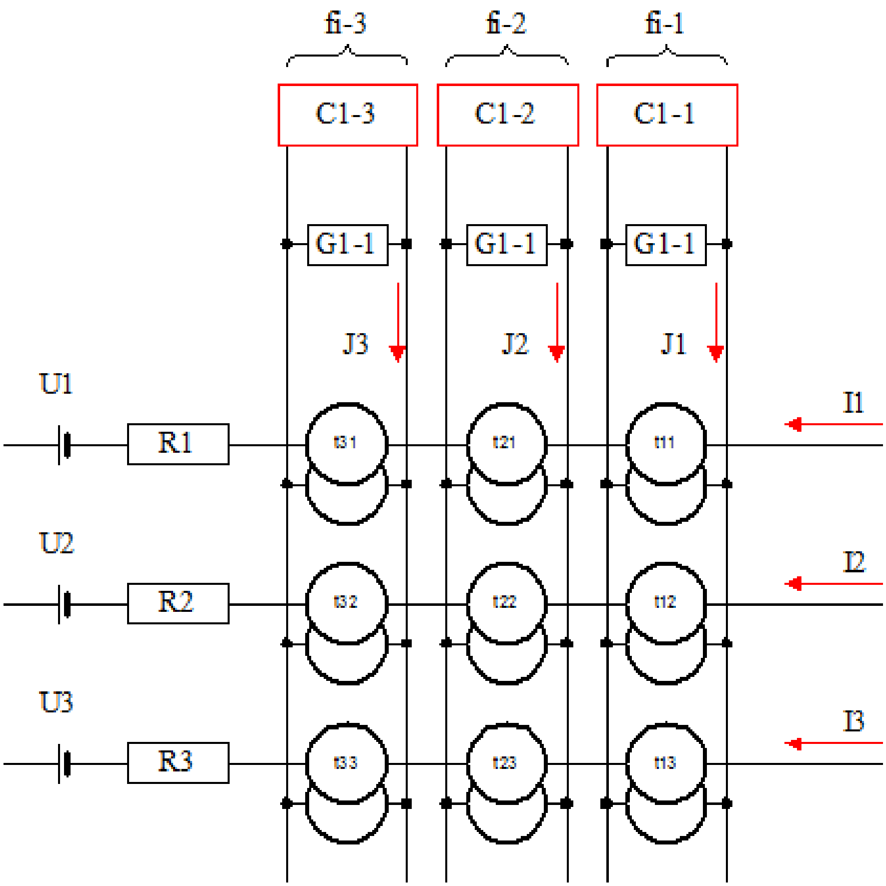



*Fig. 2. Special TD matrix.*

Let us consider now a special TD matrix – see, for instance, Fig. 2. This matrix satisfies the following assumptions:

$$J_1 = t_{11}I_1 + t_{21}I_2 + t_{31}I_3, \qquad L_1 = t_{11}\varphi_1' + t_{12}\varphi_2',$$

$$J_2 = t_{12}I_1 + t_{22}I_2 + t_{32}I_3, \qquad L_2 = t_{21}\varphi_1' + t_{22}\varphi_2',$$

$$L_3 = t_{31}\varphi_1' + t_{32}\varphi_2'.$$

In the general case let us denote:

$j$ – string number,

$k$ – column number,

$J_k$ - summary current of all windings comprising the *k*- column of the matrix,

$\varphi'_k$ - common voltage on the windings comprising the *k*- column of the matrix,

$I_j$ - the current of all windings comprising the $j$ - string of the matrix,

$e_j$ - summary voltage of all windings comprising the j - string of the matrix,

$t_{jk}$ - transformation coefficient.

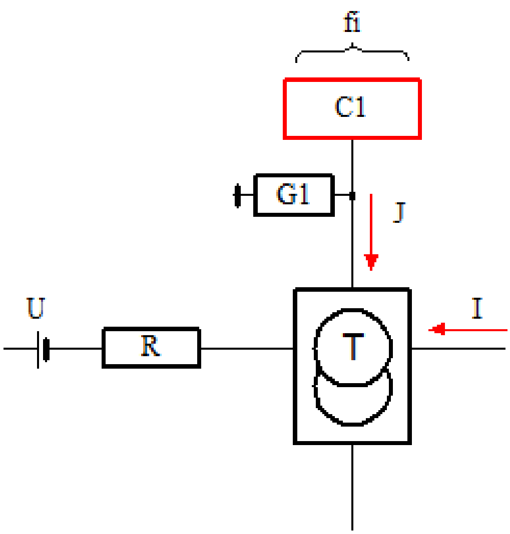



*Fig. 3. Schematic picture of electric circuit with a special TD matrix*

Schematic image of transformers matrix is given on Fig. 3. In the general case it is described by the following equations:

$$e_j = \sum_k t_{jk}\varphi'_k \,,\; J_k = \sum_j t_{jk} I_j \,.$$

Let us call *k*-column of the transformers matrix a transformer node. As in a simple circuit, an ordinary node can include a current source *C* and a conductivity *G***.** But in addition, the current source and the conductivity can also be included in a transformer node. For example,

Fig. 2 shows current sources $C'_1,\ C'_2,\ C'_3$ and conductivities $G'_1,\ G'_2,\ G'_3$. In addition to the previous paragraph we shall denote:

T – transformers matrix with marked strings and columns,
$\varphi' = \{\varphi'_k\}$ - Vector of voltages on the columns,
$i' = \{i'_k\}$ - vector of currents in the columns conductivities,
$G' = \{G'_k\}$ - vector of columns conductivities,
$C' = \{C'_k\}$ - vector of columns current sources,
$J = \{J_k\}$ - vector of columns currents,
$e = \{e_j\}$ - vector of strings voltages.

Then

$$e = T \cdot \varphi',\ \ J = T^T \cdot I \qquad (1)$$

Mind that $e \cdot I = \varphi' \cdot J$, i.e. this transformers matrix is a passive element. The first Kirchhoff law for transformer nodes looks as follows:

$$T^T \cdot I + C' = i' \qquad (2)$$

The second Kirchhoff law for transformer nodes looks as follows:

$$N^T \cdot \varphi + R \cdot I - U + T \cdot \varphi' = 0 \qquad (3)$$

Let us consider a function

$$F(I,i,i') = \frac{1}{2} \cdot I^T \cdot R \cdot I - U^T \cdot I + i \cdot \frac{1}{2G} \cdot i + i' \cdot \frac{1}{2G} \cdot i' \qquad (4)$$

Let us find the necessary conditions on this function's optimum with constraints of the form (1.2) and (2). They are similar to equations (3). Here $\varphi$ is a vector of undetermined Lagrange multipliers for the condition (1.2), which appear when the optimized function is supplemented by the term $\varphi^T \cdot (N \cdot I + C - i)$. Also here $\varphi'$ is a vector of undetermined Lagrange multipliers for the condition (2), which appear when the optimized function is supplemented by the said term $\varphi'^T \cdot \left(T^T \cdot I + C' - i'\right)$.

Further we have

$$\frac{\partial^2 F}{\partial\ I^2} = R,\ \ \frac{\partial^2 F}{\partial\ i^2} = \frac{1}{G},\ \ \frac{\partial^2 F}{\partial\ i'^2} = \frac{1}{G'}. \qquad (5)$$

It follows that the function (4) has a global minimum. Thus, the minimization of the function (4) subject to the constraints of the form (1.2) and (2) leads to equations of the second Kirchhoff' law (1).

Consequently, the calculation of the DC circuit with Dennis transformers is also equivalent to finding the conditional minimum of the function (4).

Let us now take the function

$$\Phi(\phi,\varphi)=\left\{\begin{matrix}-\phi^T\dfrac{1}{2R}\phi+C^T\cdot\varphi+C'^T\cdot\varphi' \\ -\dfrac{1}{2}\varphi^T\cdot G\cdot\varphi-\dfrac{1}{2}\varphi'^T\cdot G\cdot\varphi'\end{matrix}\right\}. \quad (6)$$

Evidently,

$$i'=G'\cdot\varphi' \quad (7)$$

and Kirchhoff law can be rewritten in the form

$$N^T\cdot\varphi+\phi-U+T\cdot\varphi'=0, \quad (8)$$

$$T^T\cdot I+C'=G'\cdot\varphi', \quad (9)$$

We find necessary conditions for the optimum of this function under constraints - equations of Kirchhoff's second law in the form (8). These conditions have the form of equations (1.10) and (9), where $I$ is a vector of undetermined Lagrange multipliers for the condition (7) that appear when the optimized function is complemented by the term $I^T\cdot\left(N^T\cdot\varphi+\phi-U+T\cdot\varphi'\right)$.

Further we have:

$$\frac{\partial^2\Phi}{\partial\ \phi^2}=\frac{-1}{R},\ \ \frac{\partial^2\Phi}{\partial\ \varphi^2}=-G,\ \ \frac{\partial^2\Phi}{\partial\ \varphi'^2}=-G'.$$

It follows that the function (6) has a global <u>maximum</u>. Thus, the maximization of (6) under the constraint in the form of second Kirchhoff law (8) leads to equations of the first Kirchhoff's laws equation (1.10), (9) and the condition (1.3). Consequently, the calculation of a DC circuit is equivalent to finding the conditional maximum of the function (6).

**Example 1. Linear programming with equality constraints.**
Let us consider a special case of an electric circuit when $N=0,\ \ R=0,\ \ \dfrac{1}{G}=0$. While calculating such circuit we in fact are solving the following problem - see (2), (4):

$$-I^T\cdot U=\min,\ \ T^T\cdot I=C'.$$

Thus there we are solving a linear programming problem with equality constraints.

Minimization of the function (4) under the constraint (3) and maximizing the function (6) subject to (7) are <u>*dual*</u>.

Equations of DC circuits with transformers Dennis **not** always have a solution. This follows from the fact that the equation (2) not always has a solution.

A circuit with "multi-winding" DT always may be transformed into a circuit with DT matrix.

**Example 2. "Multi-winding" DT.** Let us consider a circuit with "multi-winding" DT, shown on the Fig. A. The circuit shown on the Fig. B, containing the DT matrix, is equivalent to it. It becomes especially clear, if we draw it again in the form of Fig. C.

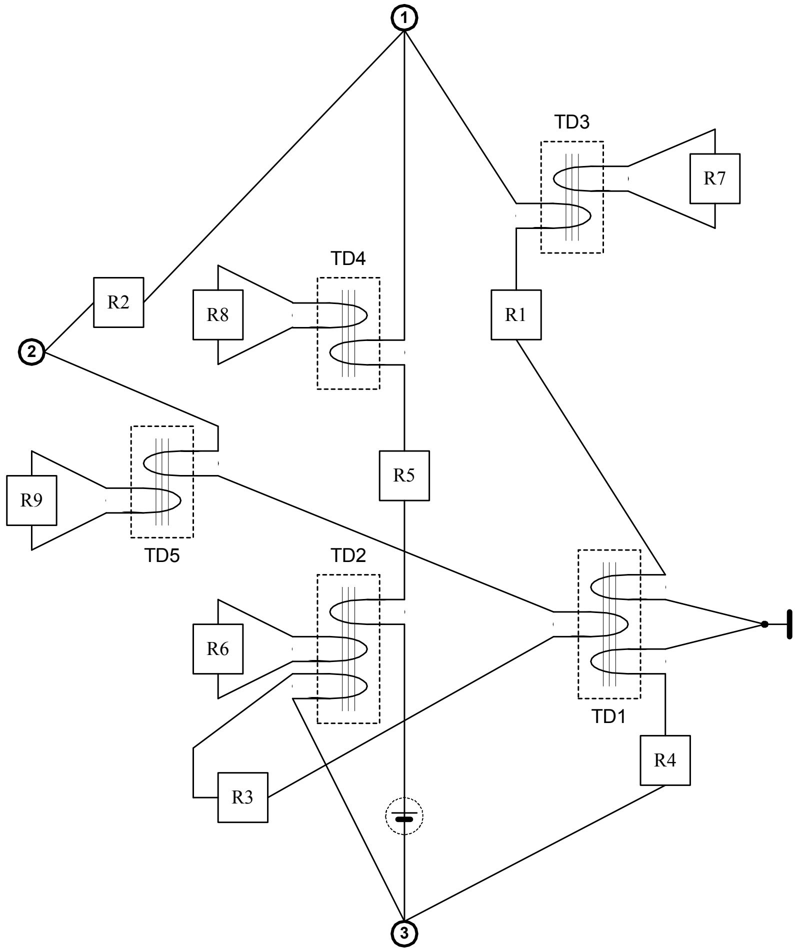



*Fig. A.*

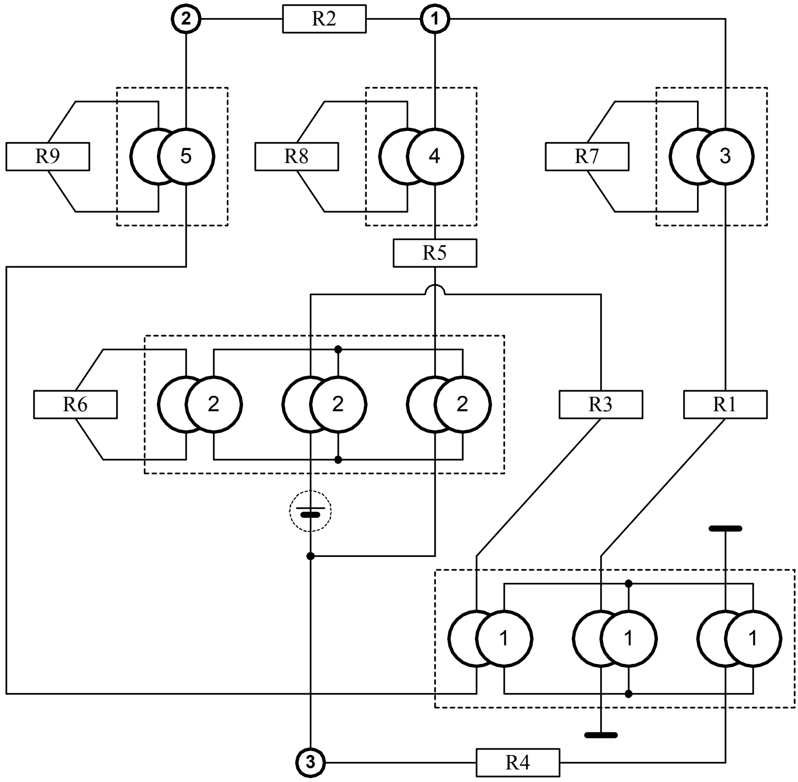



*Fig. B.*

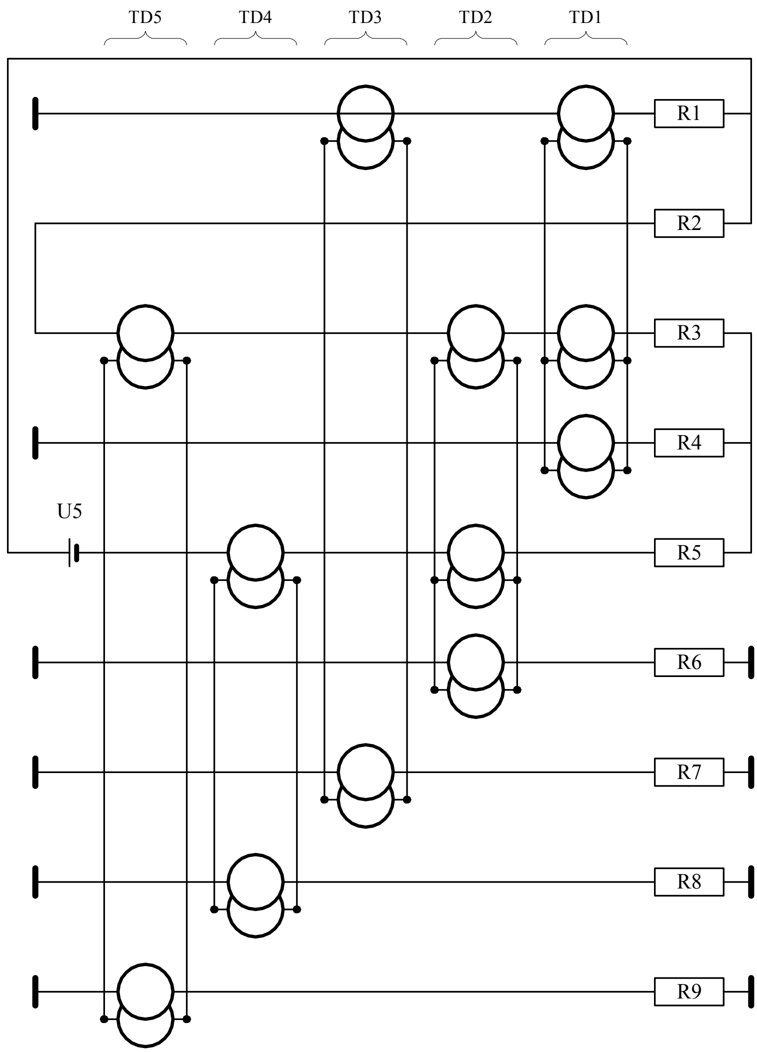



*Fig. C.*

### 1.2.2. Unconditional Electric Circuit with Dennis Transformers

We have already considered unconditional electric circuit. Here we shall supplement her with Dennis Transformers and conductivity $G' \equiv 1/\rho$, which are included between the base node and each Dennis transformer. An example of such inclusions is shown in Fig. 3, where the transformer with transformation ratio t is replaced by two transformers with transformation ratios $k_1 = t$ and $k_2 = 1$ correspondingly. Obviously, $e_1 = -\rho \cdot (i_2 + t \cdot i_1) \cdot t$ and $e_2 = -\rho \cdot (i_2 + t \cdot i_1)$. Consequently, $e_1 = t \cdot e_2$.

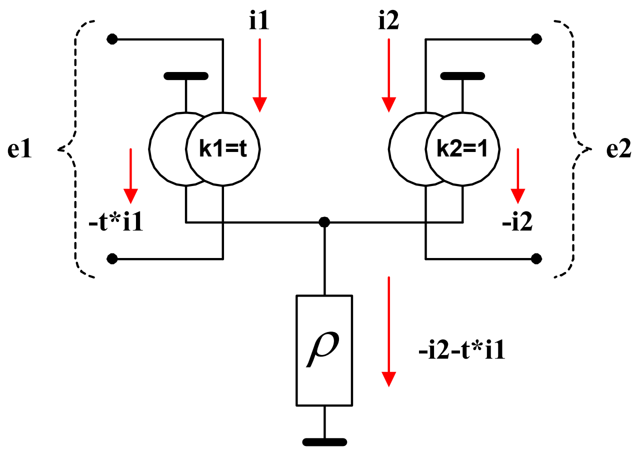



*Fig. 3. Transformer Node*

The first Kirchhoff law for ordinary nodes has the form (1.2). The first Kirchhoff law for transformer units has the form (2).

Let us consider the optimization problem of the function (4) under the constraints (1.2) and (2). In our case (4) takes the following form:

$$F(I,i,i') = \frac{1}{2} I^T \cdot R \cdot I + \frac{\rho}{2} i^T \cdot i + \frac{\rho}{2} i'^T \cdot i' - U \cdot I . \tag{10}$$

Substituting (1.2) and (2) in (10), we get

$$F(I) = \frac{1}{2} \cdot I^T \cdot R_N \cdot I - U_N^T \cdot I , \tag{11}$$

where

$$R_N = R + \rho \cdot \left( N^T \cdot N + T \cdot T^T \right), \tag{12}$$

$$U_N = U - \rho \cdot \left( N^T \cdot C + T \cdot C' \right). \tag{13}$$

Let us find necessary conditions for the unconditional minimum of the function (11). These are the equations of the second Kirchhoff law and have the form of equations

$$R_N \cdot I - U_N = 0 \tag{14}$$

or

$$R \cdot I - U + \rho \cdot \left[ N^T \cdot (N \cdot I + C) + T \cdot \left( T^T \cdot I + C' \right) \right] = 0 \tag{15}$$

The algorithm of calculating the electric circuit with DT is built in the same way as the algorithm of p.1.3.

On each iteration:

- the gradient *p* is calculated by formula (142) for the given vector *I*;
- the coefficient *a* is calculated by (1.33) for given *p*;
- the new value of vector *I* is calculated by formula (1.31).

Iterative process continues till the value

$$\varepsilon_2 = p^T \cdot p \tag{16}$$

reaches a given minimum. Virtually we should strive to the value

$$\varepsilon_2 = \varepsilon \cdot U_N^T \cdot U_N, \tag{17}$$

where $\varepsilon << 1$ is a given value of relative error. Virtually should strive to value $\varepsilon_2 = \varepsilon$.

### 1.2.3. The Transformer Connection of Lines with Nodes

Let us refer to Fig. 4, where a certain AB line with current *I* is connected to node C through a Dennis transformer with a ratio of t. In this case, a current *t*I* flows in the node C, and the potentials $\varphi$ and $\psi$ of the points A and C are connected by the relation $\varphi = t \cdot \psi$. We assume that all lines of the circuit have such *transformer connection to the nodes* (in the case of a direct connection of the line to the node, we assume that there is a connection through a transformer with t = 1), and the current sources H and nodal conductivities G are attached directly to the node. Then the first Kirchhoff law for ordinary nodes has the form (1.2), where the incidence matrix consists of elements $t_{km}$ -- transformation ratio of the km-transformer connecting the k-line with m-node. Wherein

$$\psi = \rho \cdot (N \cdot I + H) \tag{16}$$

and the potentials of the line ends and nodes are related by formula

$$\varphi = N^T \cdot \psi. \tag{17}$$

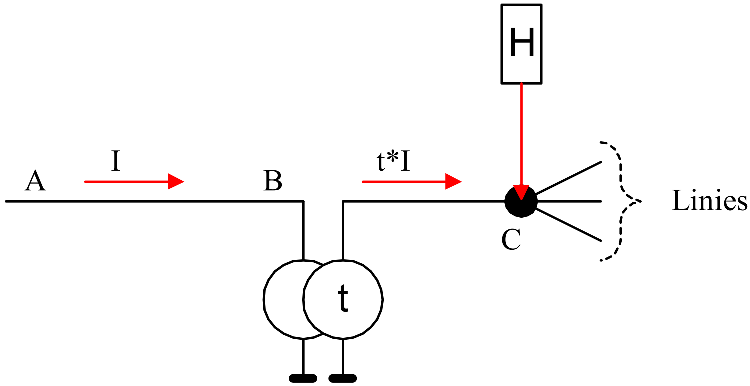



*Fig. 4. Transformer connection of line with nodes.*

**1.2.4. Unconditional electric circuit with transformer matrix**

We shall assume that in all ordinary nodes of electric circuits nodal resistors $\rho$ and current sources $C$ are included, and in all the transformer nodes, nodal transformer, resistors $\rho$ and current sources $C'$ are included. Currents flowing through the resistors $\rho$, as before, will be denoted as $i$, $i'$ for ordinary and transormers nodes accordingly. Such circuits will be called electric circuits of *ordinary type*.

We shall assume that all the ordinary nodes of electrical circuit include nodal resistors $\rho$ and current sources $C$, and all transformer nodes include nodal resistors $\rho$ and current sources $C'$. Currents flowing through the resistors $\rho$, as before, will be denoted $i$, $i'$ for ordinary and transformer nodes, respectively. These circuits will be called electric circuits of *general form*.

Fig. 5 shows an example of a circuit of general form where all nodes include node resistors and current sources. In this figure, the letters a, b, c denote branches of the transformer matrix strings and the breaks in ordinary branches where the branches of the strings are inserted. In this case the equations (1.2, 2, 10-15) are valid.

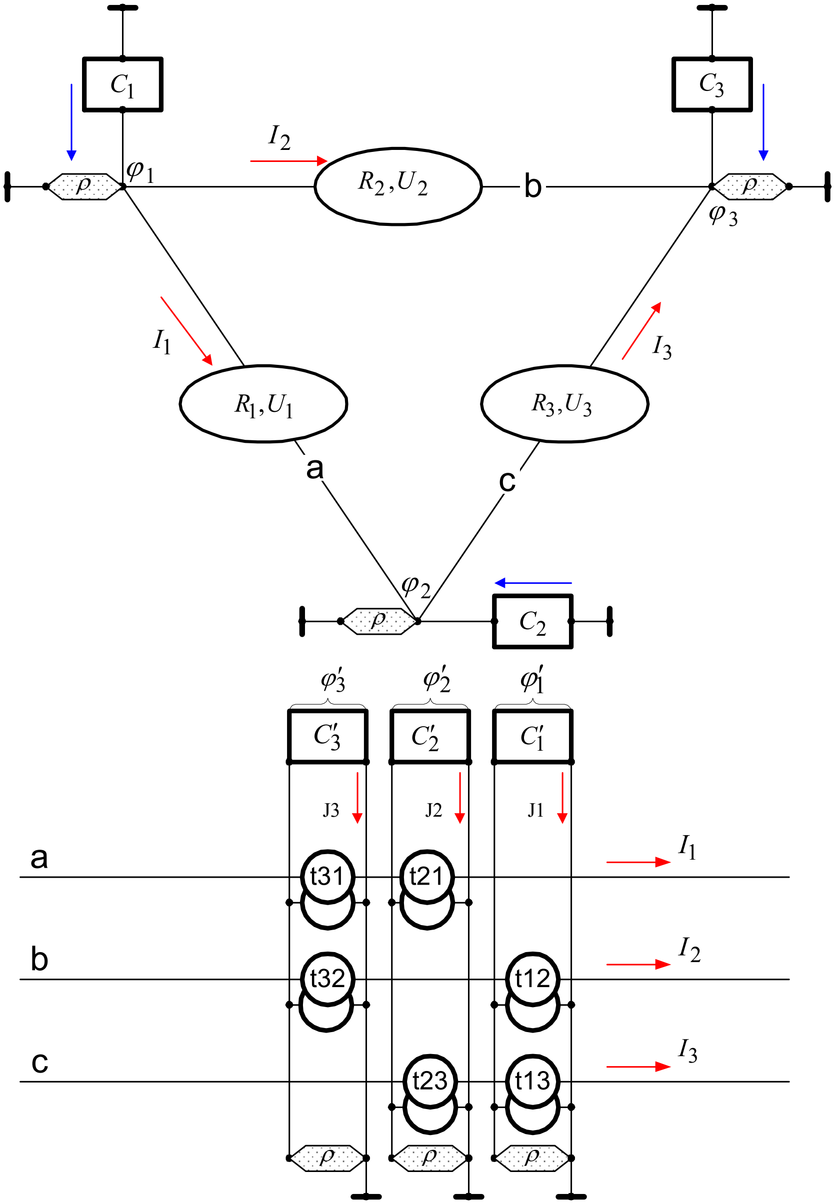



*Fig. 5. An example of an electric circuit of a general form.*

# 1.3. Special Electric circuits with Dennis Transformers

## 1.3.1. Dual DC circuits

Consider the particular case of unconditional electric circuit with nodal conductivities when $\lfloor N=0,\ C=0,\ G=0 \rfloor$ In this case in the electric circuit function of currents is being minimized

$$F(I)=\frac{1}{2}I^T\cdot R\cdot I+\frac{1}{2}i'^T\cdot(1/G')\cdot i'-U^T\cdot I\,. \tag{1}$$

under the conditions (2.2). At the same time, and the dual problem is also solved - the potentials function

$$\Phi(\phi,\varphi')=-\phi^T\cdot\frac{1}{2R}\cdot\phi+C'^T\cdot\varphi'-\frac{1}{2}\varphi'^T\cdot G'\cdot\varphi'\,. \tag{2}$$

is maximized under condition

$$T\cdot\varphi'+\phi-U=0\,. \tag{3}$$

In particular, if $G'=0$, then the primal problem takes the form:

$$\begin{aligned}&\frac{1}{2}I^T\cdot R\cdot I-U\cdot I=\min,\\&T^T\cdot I+C=0.\end{aligned} \tag{4}$$

If $R=0$, then the dual problem takes the form:

$$\begin{aligned}&-\frac{1}{2}\varphi'^T\cdot G'\cdot\varphi'+C'^T\cdot\varphi'=\max,\\&T\cdot\varphi'-U=0.\end{aligned} \tag{5}$$

It is easy to see that both problems coincide up to notations. So the electric circuit $\lfloor N=0,\ C=0,\ G=0,\ G'=0 \rfloor$ and the electric circuit $\lfloor N=0,\ C=0,\ G=0,\ R=0 \rfloor$ will be called **dual.** They are shown on Fig. 1 and Fig. 2 correspondingly.

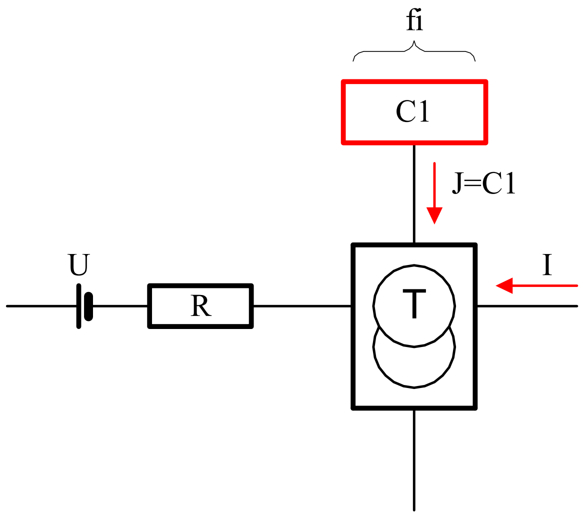



*Fig. 1. Electric circuit 1.*

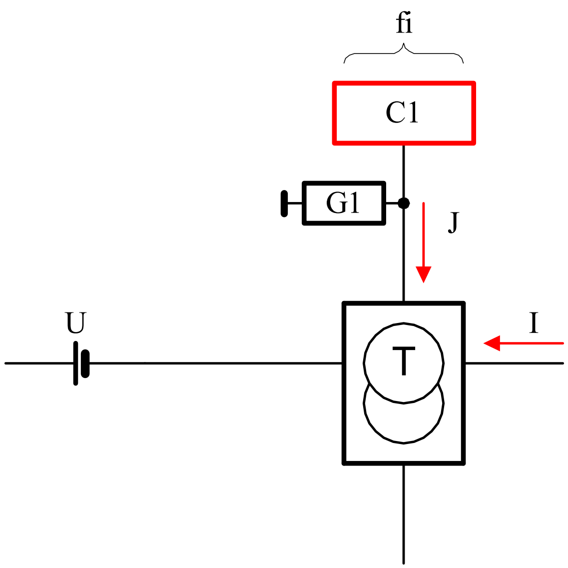



*Fig. 2. Electric circuit 2.*

### 1.3.2. Two-matrix DC Circuits

Let us consider now an electric circuit including two matrices with common branches, which will be denoted as values with one and two strokes correspondingly – see Fig. 3. In such circuit the currents are distributed in such way that the currents function

$$F(I,i,i',i'')=\frac{1}{2}I^T\cdot R\cdot I+\frac{1}{2}i^T\cdot(1/G)\cdot i$$
$$+\frac{1}{2}i'^T\cdot(1/G')\cdot i'+\frac{1}{2}i''^T\cdot(1/G'')\cdot i''-U\cdot I \tag{6}$$

is minimized under the conditions that are equations of the first Kirchhoff law (1.1.2) and

$$T'^T\cdot I+C'=i', \tag{7}$$

$$T''^T\cdot I+C''=i'', \tag{8}$$

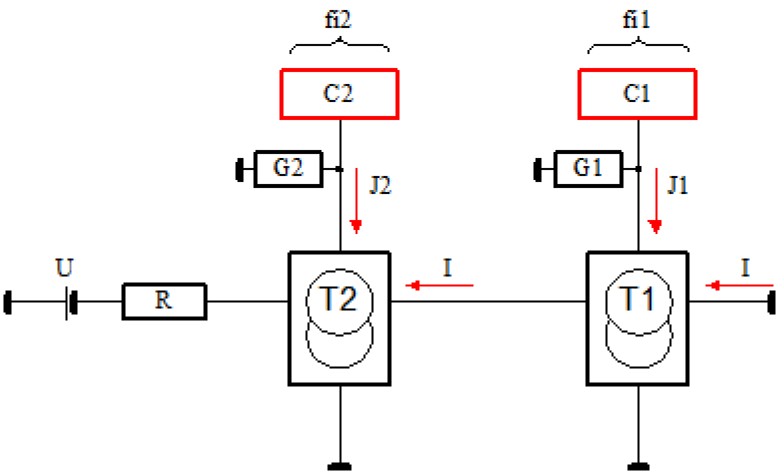



*Fig. 3. Two-matrix electric circuit*

At the same time with it the dual problem is also being solved – the potentials function

$$\Phi(\phi,\varphi,\varphi')=\left\{\begin{array}{l}-\phi^T\dfrac{1}{2R}\phi+C^T\cdot\varphi+C'^T\cdot\varphi'+C''^T\cdot\varphi''\\ -\dfrac{1}{2}\varphi^T\cdot G\cdot\varphi-\dfrac{1}{2}\varphi'^T\cdot G'\cdot\varphi'-\dfrac{1}{2}\varphi''^T\cdot G''\cdot\varphi''\end{array}\right\}. \tag{9}$$

is being maximized under the conditions that are equations of the second Kirchhoff law

$$N^T\cdot\varphi+T'\cdot\varphi'+T''\cdot\varphi''+\phi-U=0. \tag{10}$$

Let us now consider a particular case, when

$$N = 0,\ \ G = 0,\ \ G' = 0,\ \ C = 0,\ \ C'' = 0,\ \ G'' = 1.$$

Then from (8) it follows:

$$T''^{T} \cdot I = i''. \tag{11}$$

Further, from (6), (11) we find that in this case the following problem is being solved – the currents function

$$F(I) = \frac{1}{2} I^T \cdot R \cdot I + \frac{1}{2} I^T \cdot \left( T'' \cdot T''^{T} \right) I - U \cdot I \tag{12}$$

is minimized under conditions

$$T'^{T} \cdot I + C' = 0. \tag{13}$$

At the same time with it the dual problem is also being solved – the potentials function is being maximized

$$\Phi(\phi, \varphi, \varphi') = -\phi^T \frac{1}{2R} \phi + C^T \cdot \varphi + C'^{T} \cdot \varphi' - \frac{1}{2} \varphi''^{T} \cdot \varphi'' \tag{14}$$

is being maximized under the conditions that are equations of the second Kirchhoff law

$$T' \cdot \varphi' + T'' \cdot \varphi'' + \phi - U = 0. \tag{15}$$

### 1.3.3. Unconditional two-matrix circuits

We have considered above an unconditioned electric circuit with Dennis transformers. For $G \equiv 1/\rho$, $G' \equiv 1/\rho$, $G'' \equiv 1/\rho$ function (6) takes the following form:

$$F(I, i, i', i'') = \frac{1}{2} I^T R \cdot I + \frac{\rho}{2} i^T i + \frac{\rho}{2} i'^{T} i' + \frac{\rho}{2} i''^{T} i'' - U \cdot I. \tag{16}$$

For two-matrix circuit the first Kirchhoff law for ordinary nodes is expressed by the equations (1.2), (7), (8). Substituting these equations into (16), we get

$$F(I) = \frac{1}{2} \cdot I^T \cdot R_N \cdot I - U_N^T \cdot I, \tag{17}$$

where

$$R_N = R + \rho \cdot \left( N^T \cdot N + T' \cdot T'^{T} + T'' \cdot T''^{T} \right), \tag{18}$$

$$U_N = U - \rho \cdot \left( C^T \cdot N + C'^{T} \cdot T'^{T} + C''^{T} \cdot T''^{T} \right), \tag{19}$$

Necessary conditions of unconditioned minimum of function (17) have the form of equations

$$R \cdot I - U + \rho \cdot \left[ N^T (N \cdot I + C) + T' \left( T'^{T} \cdot I + C' \right) + T'' \left( T''^{T} \cdot I + C'' \right) \right] = 0. \tag{20}$$

# 1.4. Mathematical Problems in the Linear Electric Circuits

### 1.4.1. Linear Equations Systems

Consider a one-matrix unconditional circuit, where $\left[ N=0,\ C=0,\ G'\equiv\frac{1}{\rho} \right]$. The minimization problem (1.2.4, 1.2.2) takes the form:

$$\begin{aligned} &\frac{1}{2}\cdot I^T\cdot R\cdot I+\frac{\rho}{2}\cdot i'^T\cdot i'-U\cdot I=\min, \\ &T^T\cdot I-i'+C'=0. \end{aligned} \tag{1}$$

The minimization of dual problem (1.2.6, 1.2.8, 1.1.2) takes the form:

$$\begin{aligned} &\phi^T\frac{1}{2R}\phi+\frac{1}{2\rho}\cdot\varphi'^T\cdot\varphi'-C'^T\cdot\varphi'=\min, \\ &T\cdot\varphi'+\phi-U=0 \end{aligned} \tag{2}$$

**<u>Method 1.</u>** Consider the problem (1) for $U=0,\ \ R=1,\ \ \rho\to\infty$:

$$\begin{aligned} &I^T\cdot I=\min, \\ &T^T\cdot I+C'=0. \end{aligned} \tag{3}$$

This problem is equivalent to the solution of linear equations system $T^T\cdot I+C'=0$ with respect to the vector of unknowns $I$ and <u>minimization of square of the Euclidean norm of the vector of unknowns</u> $I^T\cdot I$. Thus, the calculation of such circuit is equivalent to the solution of an **underdetermined** linear equations system.

**<u>Method 2.</u>** Consider the problem (2) for $C'=0,\ \ R=0$:

$$\begin{aligned} &\varphi'^T\cdot\varphi'=\min, \\ &T\cdot\varphi'-U=0. \end{aligned} \tag{4}$$

This problem is equivalent to the solution of linear equations system $U=T\cdot\varphi'$ with respect to the vector of unknowns $\varphi'$ and <u>minimization of square of the Euclidean norm of the vector of unknowns</u> $\varphi'^T\cdot\varphi'$.

Thus, the calculation of such circuit is equivalent to the solution of an **underdetermined** linear equations system.

**Method 3.** Consider the problem (1) for $U = 0, \quad R = 0$:

$$\begin{aligned} &i'^T \cdot i' = \min, \\ &T^T \cdot I - i' + C' = 0. \end{aligned} \tag{5}$$

This problem is equivalent to the solution of linear equations system $T^T \cdot I + C' = 0.$ with respect to the vector of unknowns $I$ and minimization of square of the Euclidean norm of the vector of unknowns $i'^T \cdot i'$. Thus, the calculation of such circuit is equivalent to the solution of an **overdetermined** linear equations system.

**Example 1. Minimization of dispersion.** Consider a special case of problem (3), when the matrix $T$ is a vector of unit values, and vector $I$ is a scalar. The calculation of such circuit consist in the solution of an overdetermined linear equations system with one variable $I$, and such a value of $I$ is determined, for which the components of the given vector $C'$ have minimal dispersion.

**Method 4.** Consider the problem (2) for $C' = 0, \quad R = 1, \quad \rho \to \infty$:

$$\begin{aligned} &I^T \cdot I = \min, \\ &T \cdot \varphi' + I - U = 0. \end{aligned} \tag{5a}$$

This problem is equivalent to the solution of linear equations system $U = T \cdot \varphi'$ with respect to the vector of unknowns $\varphi'$ and minimization of square of the Euclidean norm of the vector of unknowns $I^T \cdot I$. Thus, the calculation of such circuit is equivalent to the solution of an **overdetermined** linear equations system.

**Method 5.** Consider a two-matrix circuit, where

$$\begin{bmatrix} N = 0, \quad R = 0, \quad C = 0, \quad G' = 1/\rho, \quad G = 0, \\ G'' = 1, \quad C' = 0, \quad C'' = 0, \quad T'' = 1 \end{bmatrix}$$

Then the dual minimization problem (1.3.25) and (1.3.26) takes the form

$$\begin{aligned} &\varphi''^T \cdot \varphi'' = \min, \\ &T' \cdot \varphi' + \varphi'' - U = 0. \end{aligned} \tag{6}$$

Problem (6) is equivalent to the solution of linear equations system $U = T' \cdot \varphi'$ with respect to the vector of unknowns $\varphi'$ and <u>minimization of square of the Euclidean norm of the residual</u> $\varphi''^T \cdot \varphi''$. Thus, the calculation of such circuit is equivalent to the solution of an **overdetermined** linear equations system.

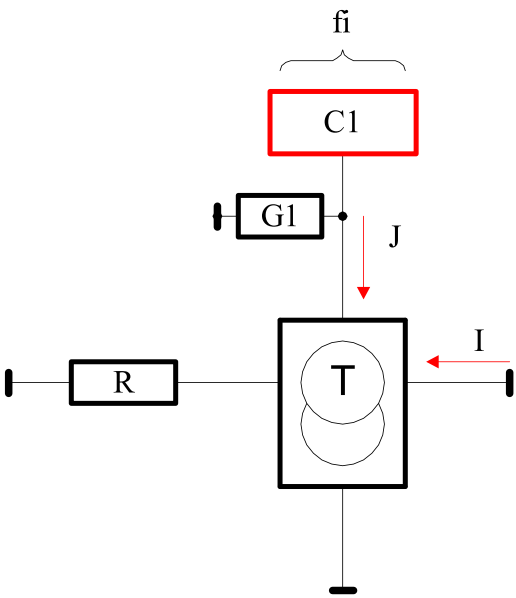
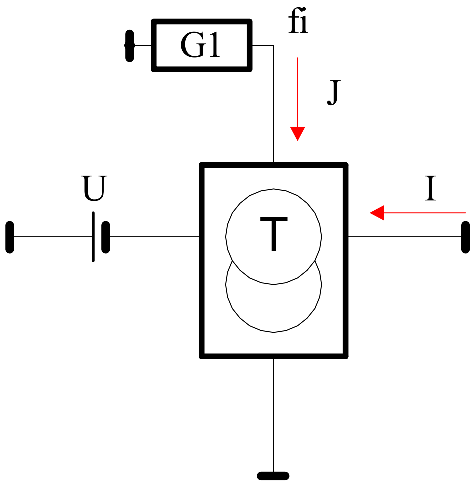
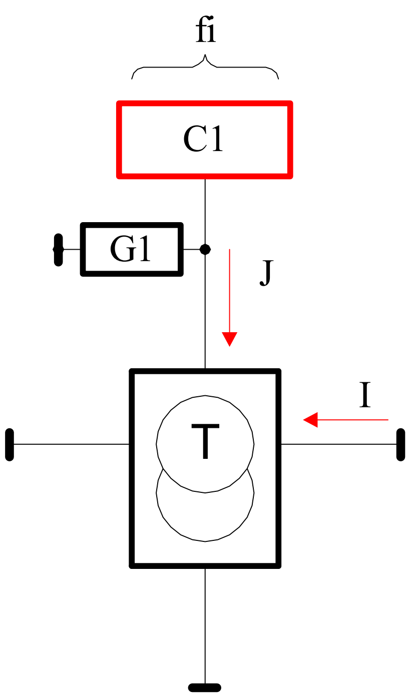


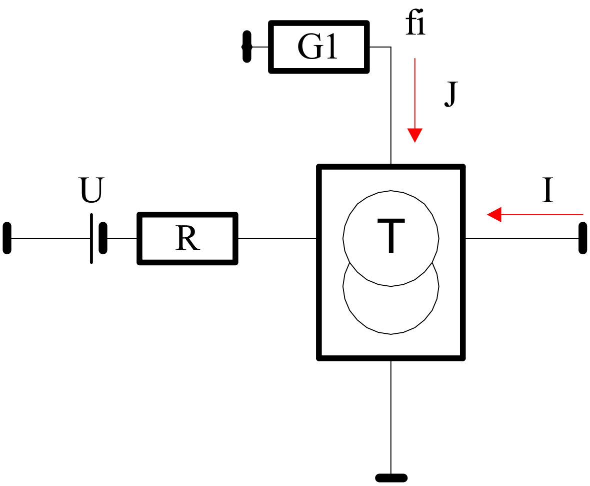

*Fig.1. To the calculation by Method 1.*



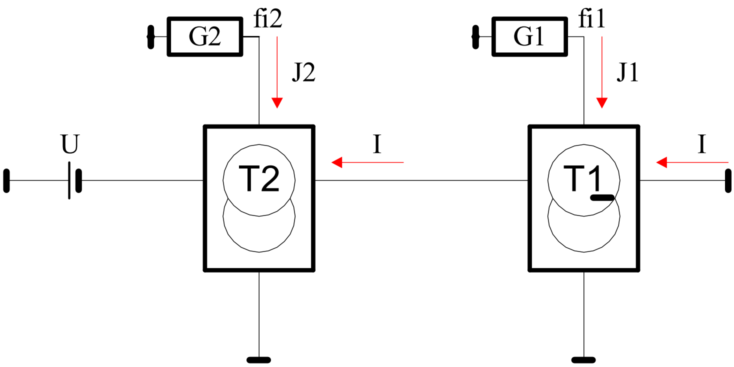

*Фиг.2. К расчету по способу 2.*




*Фиг.3. К расчету по способу 3.*




*Фиг.4. К расчету по способу 4.*




*Фиг.5. К расчету по способу 5.*

So, one-matrix circuit $\lfloor U = 0,\ \ R = 1,\ \ \rho = 0 \rfloor$ (Method 1, Fig. 1) and one-matrix circuit $\lfloor C' = 0,\ \ R = 0 \rfloor$ (Method 2, Fig. 2) are solving an **underdetermined** linear equations system. One-matrix circuit $\lfloor U = 0,\ \ R = 0 \rfloor$ (Method 3, Fig. 3), one-matrix circuit $\lfloor C' = 0,\ \ R = 1 \rfloor$ (Method 4, Fig. 4) and special two-matrix circuit (Method 5, Fig. 5) are solving one and the same **overdetermined** linear equations system.

For solving the well-**determined** linear equations system any of above shown methods can be used.

### 1.4.2. Program for Solution of Linear Equations System

Let us use Method 3 and let's rewrite (5.1.5) in the form:

$$T^T \cdot x + C - n_1 = 0, \quad (1)$$

$$n_1{}^T \cdot n_1 = \min. \quad (2)$$

The system (1) may be incompatible, underdetermined or overdetermined. We search for a solution corresponding to minimum (2). The M-function for solving this problem is:

```
function [x,eps1,k,n1,mini]=...
    anySLAE2(A,B,r,eps2,kmax)
```

**A** – the matrix $A = T'$ – see (1),

**B** – the vector $(-C)$– see (1),

**r** – the value $\rho$,

**eps2** – the value $\varepsilon_2$,

**kmax** – maximal iterations number,

The output values here are:

**x** – the vector $I$ – see (2.4, 1),

**eps1** – the relative residual value $\varepsilon_1$,

**k** – iterations number,

**n1** – the residuals vector in equation (1),

**mini** –value of the minimum (2).

The M-functions for test problems looks as:

```
function test_anySLAE2()
```

The test includes the solution for various types of systems. For control the same system is being solved by MATLAB means. Here the values similar to the output values of the function **anySLAE2** are computed by traditional methods. Parameter **mode** defines the test's number. Let us consider these tests.

1. Well determined small dimension system.
2. Underdetermined small dimension system,
3. Overdetermined small dimension system,
4. Overdetermined system for the computation of such vector **x**, with respect to which the components of vector **B** have minimal dispersion.
5. Underdetermined large dimension problem.
6. Overdetermined large dimension problem.
7. Poorly determined small dimension problem. In this case the problem cannot be solved by MATLAB means: a message "Warning: Matrix is singular to working precision" is displayed.

**1.4.3. Quadratic programming with equality constraints**

Consider a special case of two-matrix circuit; when the two-matrix scheme has the following form

$$\left[N=0,\;\; G=0,\;\; G'=0,\;\; G''=1,\;\; C=0,\;\; C''=0,\;\; R=0\right]$$

In this case the in the process of calculating the electrical circuit we will be solving the following problem: the function of currents should be minimized – see (1.3.28) and (1.3.29):

$$\Phi=\frac{1}{2}I^T\cdot\left(T''^T\cdot T''\right)I-U\cdot I \tag{7}$$

under the condition

$$T'^T\cdot I+C'=0 \tag{8}$$

If the matrix $A=\left(T''^T\cdot T''\right)$ is a square one, then the problem is reduced to quadratic programming problem in its traditional form:

$$\begin{gathered}\frac{1}{2}I^T\cdot A\cdot I-U\cdot I=\min,\\ T'^T\cdot I+C'=0.\end{gathered} \tag{9}$$

Note that the solution of this problem exists if the matrix $A$ is positive definite. The appropriate scheme is depicted on Fig. 6.

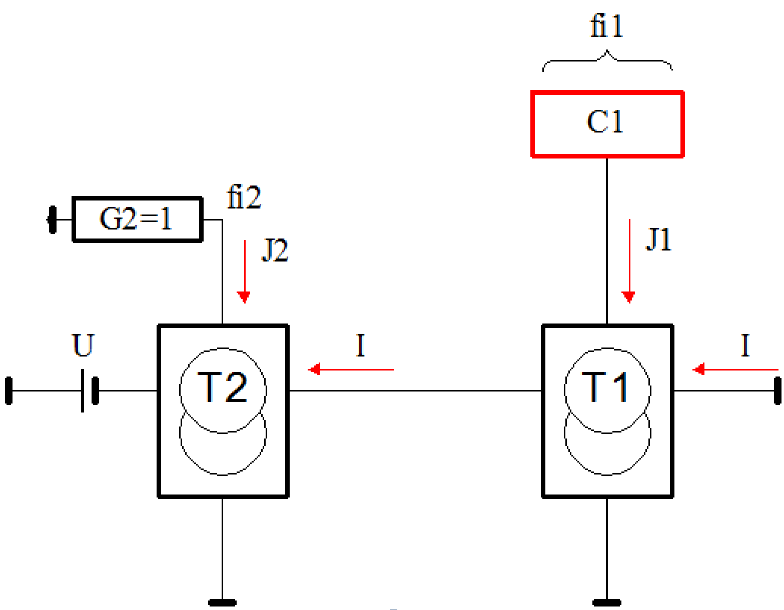



*Fig. 5. To the quadratic programming problem*

The solution method is associated with the transformation $A = T''^{T} \cdot T''$, resulting in definition of matrix $T''$. In the case when matrix $A$ is symmetrical, for such transformation it is possible to use the decomposition of LU-matrix on two triangular matrices $U = T''$ и $L = T''^{T}$. The method of such decomposition is known [38] and will not be described here. As the result of such decomposition, the quadratic programming problem (23) is also reduced to two one-matrix circuits' calculations.

# Chapter 2. Nonlinear DC Circuits

## 2.1. Electric circuit with diodes

### 2.1.1. Method of calculation circuits with diodes

For circuits containing resistors, diodes and Dennis transformers in [2] it is shown that the DC voltages and currents in these circuits are the solution of a *quadratic programming problem with inequality constraints*. More precisely, the minimization of the function (1.2.4) under the constraints of the form (1.1.2), (1.2.2) and $I_d \geq 0$ leads to the equations of the second Kirchhoff's law (1.2.1), the equations $U_d \geq 0$ and the complementary slackness condition $I_d \cdot U_d = 0$. Here $I_d$, $U_d$ are the currents and voltages of the diodes. Next, we consider another approach in which the diode is replaced (approximately) the nonlinear resistance that allows to replace the specified problem by a *convex programming problem without restrictions.*

By analogy with the previous discussion let us consider an unconditioned electric circuit where the minimized function is

$$F(I) = \frac{1}{2} I^T R \cdot I - I^T E + \frac{\rho}{2} (N \cdot I + H)^T (N \cdot I + H), \tag{1}$$

where $R$ is a nonlinear resistor with non-decreasing voltage-current characteristic $\varphi(I)$. Let us denote

$$\mu(I) = \frac{1}{2} I^T R \cdot I \tag{2}$$

and rewrite (1) as

$$F(I) = \mu(I) - I^T E + \frac{\rho}{2} (N \cdot I + H)^T (N \cdot I + H). \tag{3}$$

The gradient of currents vector $I$ is

$$p = \frac{d\mu(I)}{dI} - E + \rho N^T (N \cdot I + H). \tag{4}$$

For the given values of vector $I$ its new value is calculated by the formula

$$I_n = I - a \cdot p\,. \tag{5}$$

When the vector changes from $I$ to $I_n$ the function (3) changes by $\Delta F(a) = F(I_n) - F(I)$. Next we have

$$\frac{d\Delta F}{da} = \frac{dF(I_n)}{da} = \frac{dF(I_n)}{dI_n} \cdot \frac{dI_n}{da} = -p^T \frac{dF(I_n)}{dI_n}.$$

Thus,

$$\frac{d\Delta F}{da} = -p^T \left( \frac{d\mu(I_n)}{dI_n} + \rho N^T (NI + H) - E \right).$$

The optimal value of the step $a$ is determined from $\frac{d\Delta F}{da} = 0$ [3]:

$$A = -p^T \cdot \left( \frac{d\mu(I)}{dI} + RI + \rho N^T (NI + H) - E \right), \tag{6}$$

$$B = -p^T \left( \frac{d^2\mu(I)}{dI^2} + R + \rho N^T N \right) p. \tag{7}$$

Confining ourselves to first-degree polynomial, we find:

$$a \approx \frac{-A}{B}. \tag{8}$$

For linear resistors $\frac{d\mu(I)}{dI} = RI$, $\frac{d^2\mu(I)}{dI^2} = R$. In other cases these dependencies become more complicated. Consider for instance the case when

$$\mu(I) = \ln(1+I) - 1 + I\ln(1+I) - I. \tag{9}$$

Then

$$\frac{d\mu(I)}{dI} = \ln(1+I), \tag{10}$$

$$\frac{d^2\mu(I)}{dI^2} = \frac{1}{I+1}. \tag{11}$$

In particular, we have

$$\left.\begin{aligned}
&\frac{d\mu(I)}{dI}=0, \ \text{ if } \ I=0;\\
&\frac{d\mu(I)}{dI}=\infty, \ \text{ if } \ I=-1;\\
&\frac{d\mu(I)}{dI}=-1, \ \text{ if } \ I=\frac{1}{e}-1;\\
&\frac{d\mu(I)}{dI}=1, \ \text{ if } \ I=e-1.
\end{aligned}\right\} \qquad (12)$$

The function $\frac{d\mu(I)}{dI}$ is the volt-ampere characteristic (10) of the resistor included in the appropriate branch. Fig. 1 shows the linear resistor R, the diode D, quasi-diode qD, a source of constant power P and shows their current-voltage characteristics E = E (*I*) and the dependence of the resistance on the current R = R (*I*). Quasi-diode qD has a voltage-current characteristic (10). It reminds voltage-current characteristic of the diode – see also (12). With this in mind, consider the electric circuit with diodes. There we shall by represent quasi-diodes with diodes with nonlinear resistance. Minimization of the function (3) corresponds to the calculation.

$$D(I)=\frac{1}{I+1}. \qquad (13)$$

In such a circuit is minimized function (3), which $\mu(I)$ is defined by (2) for the branches with a linear resistance, and - according to (9) for the branches with diodes. Minimization of function (3) corresponds to the calculation circuits with diodes.

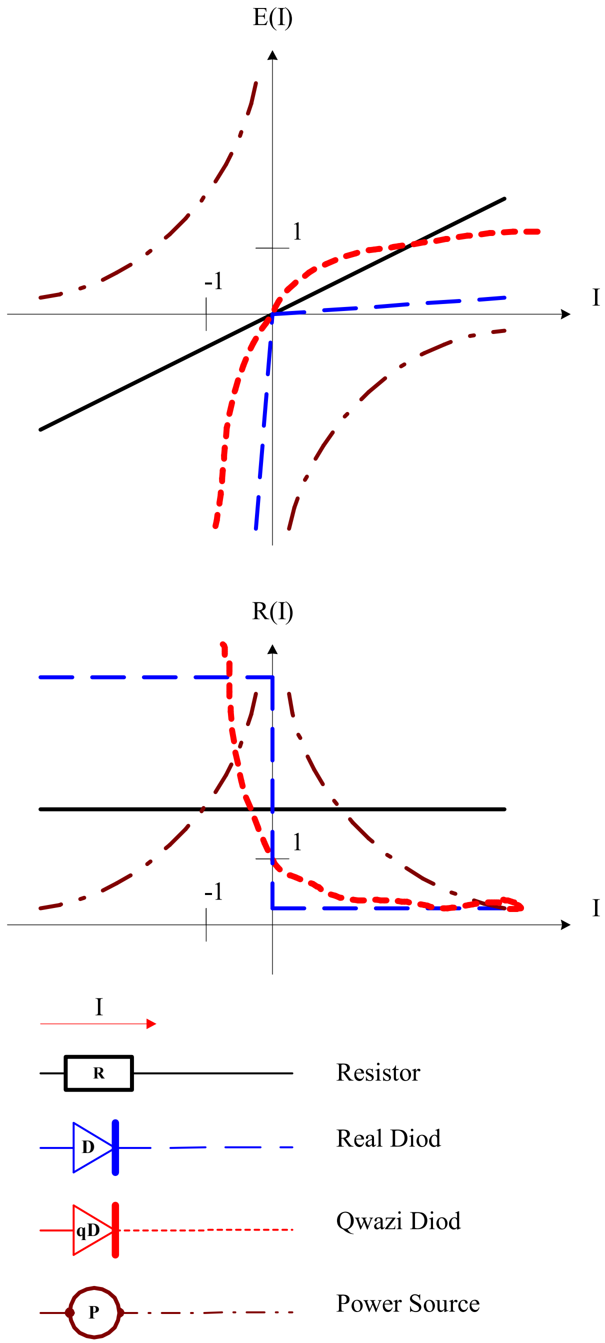



*Fig 1. Volt-ampere characteristics.*

### 2.1.2. Program for Calculation of electric direct current circuits with diodes

#### 2.1.2.1. Introduction

Below we describe a program of calculating an electric circuit with diodes in MATLAB system.

#### 2.1.2.2. Description of electric circuit

Initially, the electric circuit nodes are numbered in random order. Branches electric circuits are determined by the numbers **N1** initial and **N2** end node. A branch may contain a resistance **R**, a constant voltage source **U** and a diode **D**. The choice of which of the two nodes to designate initial, has value only if the branch contains a diode. The initial node is assigned one that is adjacent to the positive pole of the diode. Current (defined as a result of the calculation) has a positive direction from the initial to the final node.

Description of the electric circuit is an array of **B**. Each row of the array describes one of the branches and has the following form:

```
B(k,:)=[N1,N2,R,U,D]
```

In this case, **D=1**, if the diode is in the branch, and **D=0** otherwise.

In addition, the electric circuit may include current sources, are included among the total points and some node. The positive direction of current from the source is sending to the node.

Description of current sources is an array following form:

```
C=[C1,C2,...,CN,...],
```

where each node **N** we associate the number of **CN** - the current value of the current source, or zero if in this node is not a current source.

#### 2.1.2.3. On the nodal current sources

In a real electric circuit current source is included in some branch. To bring this electric circuit to the "canonical" form (described above), do the following:

o converts the source of the current in a separate branch (not containing other elements)

o convert a branch in the two current sources, as in the electric circuit shown in Fig. 1 - see the transformation **a--->b.**

Obviously, the canonical electric circuit must satisfy the condition

```
sum(C)=0.
```

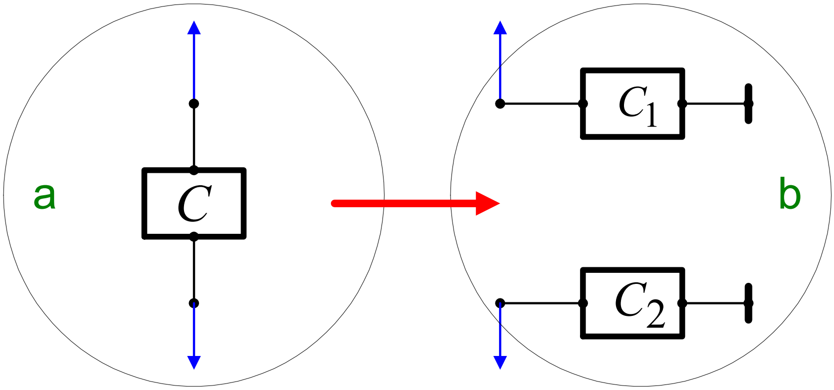



Fig. 1.

**2.1.2.4. Calculation electric circuit**

The calculation electric circuit is performed iteratively and the result of calculation, as a rule, is approximate. In this case, due to appear residuals - the deviation from zero in the equations of Kirchhoff's laws. They correspond to the relative error violations of these laws. The relative error violations the first law of Kirchhoff defined as the ratio of the mean square residuals of the first Kirchhoff's law to the mean square of the currents in the branches, and the relative error violations of the second law of Kirchhoff defined as the ratio of the mean square residuals for the second Kirchhoff's law to the mean square of the voltage on the branches (created voltage sources and current sources).

The value of the permissible relative error in violation of the second Kirchhoff's law is given by the user.

The number of iterations (i.e. duration of the calculation) and the error performance of the first law Kirchhoff adjust the amount of so-called "methodical" resistance. It makes sense the resistance included between each node and a common point. This resistance must be much greater than all the resistance branches (not counting reverse resistance diodes). The greater this resistance, the higher the accuracy of compliance of the first law of Kirchhoff, but the longer the duration of the calculations.

**2.1.2.5. Programm Description**

M-function for the calculation is as follows:

```
function [i,f,er1,er2,k,p,E,N,y,m]=...
   rucd(B,C,r,erd,dmin,dmax,n)
```

Input arguments here are

**B** –array of branches (described above),

**C** – array of current sources (described above),

**r** – "methodical" resistance,

**erd** – permissible given the magnitude of relative error violations of the second law of Kirchhoff,

**dmin** – resistance of the diode for direct current,

**dmax** - resistance of the diode for reverse current,

**n** - number of nodes.

**kmax** – allowable number of iterations..

Output values here are

**i** –an array of current of branches

**f** –the array of nodal potentials

**er1** – relative error violations the first law of Kirchhoff,

**er2** – relative error violations of the second law of Kirchhoff,

**k** – the number of iterations

**p** –the array of residuals in the branches of the second law of Kirchhoff,

**E** – the array of potential difference between the nodes of branches,

**N** – the array - incidence matrix,

**y** – the array of residuals in the nodes of the first law of Kirchhoff,

**m** – a flag of the result, where

- **m=0**, if the calculation is made;
- **m=1**, if the calculation is not carried out due to violations of conditions of **sum(C)=0;** then get a message **msg=sum(C);**
- **m=2**, if in table nodes met the node number, exceeding a specified number of nodes; then get a message is **msg='greatest number'.**

### 2.1.2.6. Example of a "canonical" form

Fig. 2 shows a simple electric circuit with diodes. Here **r** – methodical resistance, which is absent in the real electric circuit. For this electric circuit these arrays are as follows:

```
B=[1,2,R1,U1,D1;...
   1,3,R2,U2,D2;...
   2,3,R3,U3,D3];
C=[C1,C2,C3]';
```

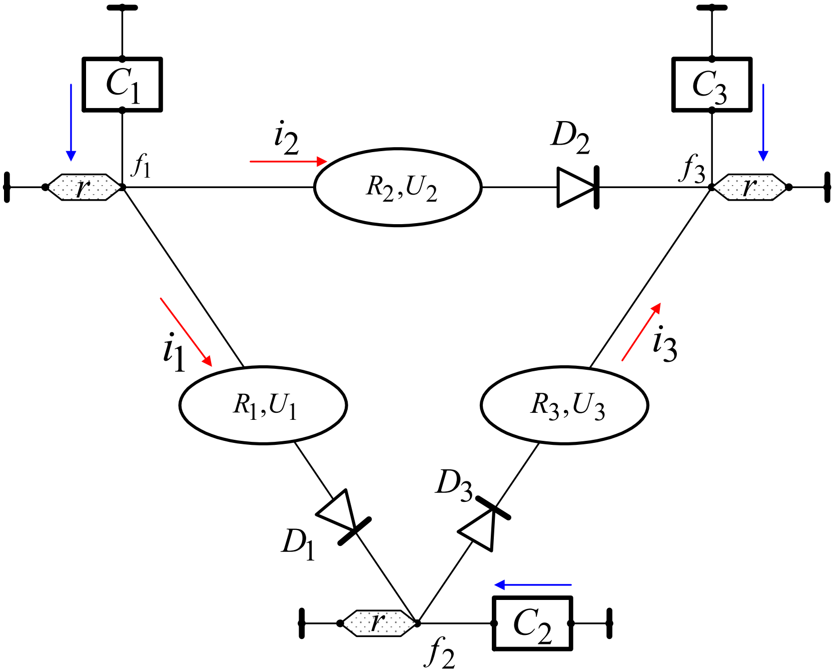



Fig. 2.

**2.1.2.7. Test**

Test M-function

```
function test()
```

calculates the electric circuit shown in Fig. 3.

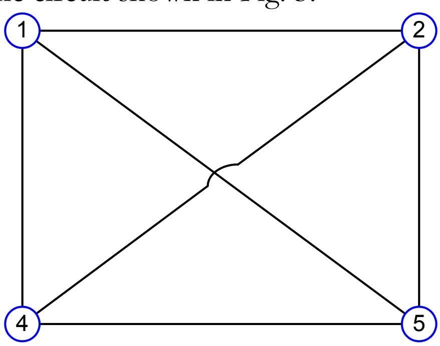



Fig. 3.

The presence of certain elements of the branches and sources of current determined in the arrays **B,C**.

# 2.2. Mathematical Problems in Electric Circuits with Diodes

### 2.2.1. Introduction

As well as for the linear electric circuits, methods of calculation of electric circuits with diodes can be used as methods for solving various mathematical problems. The following are descriptions of the respective programs in the MATLAB system. Note again that this the us of this approach, in which the diode is replaced with (approximately) by nonlinear resistance, allows us <u>to replace the problem with constraints by a convex programming problem without constraints</u>

### 2.2.2. Program for Solution of Linear Equations and Inequalities System

The system to solve is as follows:

$$T \cdot x - U \geq 0. \tag{1}$$

System (1) may be incompatible, underdetermined, overdetermined. We are searching for such solution, that corresponds to the minimum

$$x^T \cdot x = \min. \tag{2}$$

The M-function for solving this problem has the form:

```
function [k,x,er2K,n2eq,n2neq,minf]...
   =anySLAE3(A,B,D,r,eps2,kmax)
```

<u>The input</u> arguments here are:

**A** – is the matrix $A = T$ – see (1),

**B** – vector $(U)$– see (1),

**D** – the equation type indicators vector: if the m-equation is an equality, then $D_m = 0$, in the opposite case $D_m = 1$,

**r** – the value $\rho$,

**eps2** – the value $\varepsilon_2$,

**kmax** – maximal iterations number.

<u>The output</u> arguments here are:

**k** – iterations number,

**x** – vector potentials $\varphi$– see (2),

**ep2K** – the reached value $\varepsilon_2$,

**n2eq** – vector of residuals in the equalities of (1),

**n2neq** – vector of residuals in the inequalities of the (1), which by the problem's conditions may have any positive value.

**minf** – the minimum value (2).

The M-function for test problems is:

```
function testqw3
```

The test may solve various versions of the system. In some cases for the sake of control the same system is solved by MATLAB means. The parameter **`eq`** determines the number of such solution method In the test parameter **`mode`** determines the test number. Let us consider these tests.

1. Well determined small dimension system. The control is performed according to formula **`y=A\B.`**
2. Underdetermined small dimension system. The control is performed according to formula **`y=A\B.`**
3. Overdetermined small dimension system. The control is performed according to formula **`y=A\B.`**
4. Well determined equalities system **`(A,B)`**. The control is performed according to formula **`y=A\B.`**
5. The inequalities system **`(A,B)`**, coinciding in its left parts with the equalities of the system from p. 4. This system cannot be solved by MATLAB means as a quadratic programming problem A message `"No active inequalities"` is displayed.
6. The inequalities system **`(A,B)`**, coinciding in its left parts with the equalities of the system from p. 4. The equality and inequality signs may be determined by the user in the vector **`D`**. The control of this problem by MATLAB means is not performed.
7. Underdetermined inequalities system. When trying to solve it by MATLAB means as a quadratic programming problem a message `"A must have 2 column(s)"` is displayed.
8. Overdetermined inequalities system. When trying to solve it by MATLAB means as a quadratic programming problem a message `"A must have 6 column(s)"` is displayed.
9. Overdetermined system of equalities and inequalities (in its left part the same as in p. 8). The vector **`D`** is such that the obtained solution has large residuals in the equalities.
10. Overdetermined system of equalities and inequalities (in its left part the same as in p. 8). The vector **`D`** is such that the obtained solution has large residuals in the equalities.

### 2.2.3. Program for calculation of DC circuits with diodes and instantaneous values transformers

Below we describe this program in MATLAB system. The simulating scheme includes:

- nodes,
- branches,
- current sources,
- DT transformers,
- Current sources for DT.

The description of arrays B and C, as well as the description of nodal current sources was given above.

Description of the set DT represents matrix in which the rows and columns are indicated. Each element of the matrix represents one of the transformation coefficients $t_{jk}$. Each line combines DTs whose "primary windings" are connected in series and included in a certain branch. Each column of the matrix incorporates DTs, in which the "secondary windings" are connected in parallel. In series with such column a current source **CtN** can be included.

The description of current sources **CtN** presents a specific array of the following form:

```
Ct=[Ct1,Ct2,...,CtN,...].
```

Each node **N** is associated with an appropriate **CtN** – the value of current in the current source, or zero if in those DT columns where there is no current source.

The scheme calculation is performed iteratively, as described above.

M-function for the calculation is as follows:

```
function [i,f,er1,er2,k,p,E,N,yN,...
  ft,yti,ert,dD,ytf,nuz,m,ki,ro]=...
  rucd3(B,C,r,erd,dmin,dmax,n,T,Ct,...
  kmax,eri,kimax,io)
```

<u>Input</u> <u>arguments</u> here are

**B** –array of branches (described above),

**C** – array of current sources (described above),

**r** – "methodical" resistance,

**erd** – permissible given the magnitude of relative error violations of the second law of Kirchhoff,

**dmin** – resistance of the diode for direct current,

**dmax** - resistance of the diode for reverse current,
**n** - number of nodes.
**T** – DT array,
**Ct** – array of current sources in DT columns,
**kmax** – permissible number of iterations,
**eri** –given value of permissible relative error for violation of the first Kirchhoff law,
**kimax -** permissible number of cycles for increasing the "methodical" resistance,
**io** – initial value of currents in branches.

Output values here are

**i** –an array of current of branches
**f** –the array of nodal potentials
**er1** – relative error violations the first law of Kirchhoff,
**er2** – relative error violations of the second law of Kirchhoff,
**k** – the number of iterations
**p** –the array of residuals in the branches of the second law of Kirchhoff,
**E** – the array of potential difference between the nodes of branches,
**N** – the array - incidence matrix,
**yN** - array of residuals in the nodes according to the first Kirchhoff law,
**ft** – array of voltages on the DT columns,
**yti** – array of residuals on the DT columns according to the first Kirchhoff law,
**ert** – array of residuals in the branches according to the second Kirchhoff's law,
**dD** – array of diode resistors (equal to the maximum or minimum value),
**ytf -** array of voltages on the DT lines,
**nuz** – the found number of nodes (see message below **m=2**),
**ki** – number of cycles for increasing the "methodical" resistance
**ro** – the final value of "methodical" resistance
**m** – a flag of the result, where

- **m=0**, if the calculation is made;

- **`m=1`**, if the calculation is not carried out due to violations of conditions of **`sum(C)=0;`** then get a message **`msg=sum(C);`**
- **`m=2`**, if in table nodes met the node number, exceeding a specified number of nodes; then get a message is **`msg='greatest number'.`**

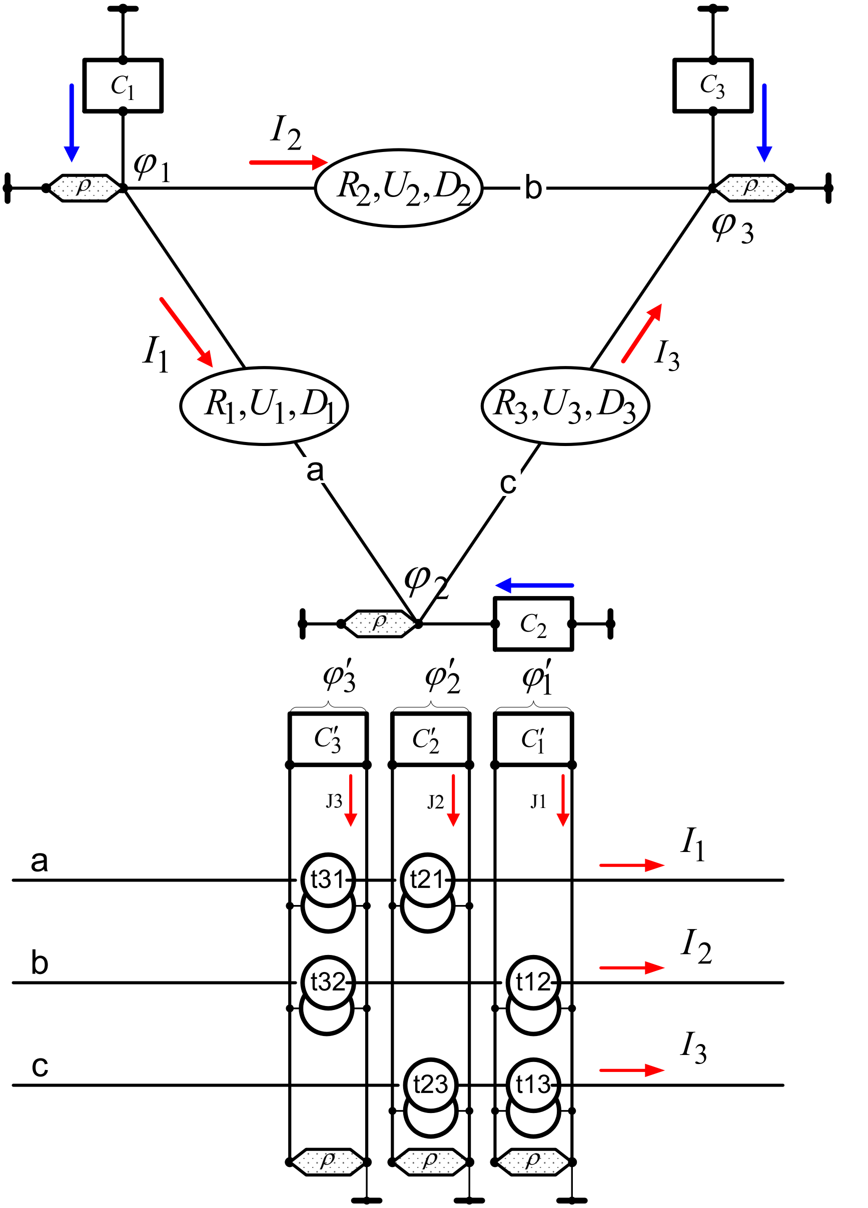



Fig. 2.

Fig. 2 presents a simple electrical circuit. Here **r** is the methodical resistance that is absent in the actual circuit. For this circuit, these arrays are as follows:

```
B=[1,2,R1,U1,D1;...
   1,3,R2,U2,D2;...
   2,3,R3,U3,D3];
T=[t31,t21,t11;
   t32,t22,t12;
   t33,t23,t13];
C=[C1,C2,C3]';
Ct=[Ct1,Ct2,Ct3]';
```

The test M-function `function testrucd3()` performs the calculation of the circuit shown on Fig. 2.

#### 2.2.4. Quadratic Programming

The following describes the program for solving Quadratic Programming Problems in MATLAB.

##### 2.2.4.1. The First Problem of Quadratic Programming

Consider the following problem.. To find minimum of the function

$$F(x) = 0.5 \cdot x^T \cdot R \cdot x - U^T \cdot x \tag{1}$$

under the restraints

$$T^T \cdot x + C = 0, \tag{2}$$

$$x \geq 0, \tag{3}$$

where

$T$ is a superscript - the transposition sign,

$x, C, U$ - $k$-dimensional vectors,

$T$ - $k \cdot m$ matrix

$R$ - a $k \cdot k$ square positive definite matrix

The unknown variable here is vector $x$. The equations set (2) must not be empty. The constraint (3) may relate only to certain variables, or not exist at all. Further in this problem we shall use vector **D** – the equation type indicator in (3): if an m-equation is an equality, then $D_m = 0$, in an opposite case $D_m = 1$.

The solution method is based on movement along the gradient of the minimized function (1). The computation is going in iterations, and the result as a rule is approximate. As a result there appear residuals in the equations (2, 3). The relative residual in the equation (2) is determined as

$$\varepsilon = \left(T^T \cdot x + C\right)^T \left(T^T \cdot x + C\right) / \left(x^T x\right) \tag{4}$$

The permissible value of this residual $\varepsilon_{\max}$ is given by the user.

The number of iterations (i.e. the computation time) and the value $\varepsilon_{\max}$ are regulated by the value $\rho$. The larger is the value $\rho$, the less is the value $\varepsilon_{\max}$, but the longer is computation time.

The M-functions for this problem solution is:

```
function [x,n1,k,ero,Fmin,...
              kk,erok,Fmink,xx]=...
squ2 (R,T,C,U,D,r,erd,kmax)
```

<u>The input</u> arguments here are:

**R,T,C,U, D** – matrices and vectors defined in (1, 2),

**r** – the value $\rho$,

**erd -** the value $\varepsilon_{\max}$,

**kmax** – maximal iterations number,

<u>The output</u> arguments here are:

**x** – the unknown vector,

**n1** – the vector of residuals in the equations (2),

**k** – iterations number,

**ero** – value of relative residual $\varepsilon$,

**Fmin** – minimum of the function (1).

The following output values are used for creating the graphs:

**kk** – the vectors of iterations numbers,

**erok** - the vector of relative residuals $\varepsilon$ in each iteration,

**Fmink** – the vector of function (1) minimum value on each iteration,

**xx** - the matrix of **x** vectors on each iteration.

### 2.2.4.2. The Second Problem of Quadratic Programming

Let us consider now the following problem. We are searching for the minimum of function (1)

$$F_1(x) = 0.5 \cdot x_1^T \cdot R \cdot x_1 - U_1^T \cdot x_1 \tag{5}$$

under the restraints

$$T_1^T \cdot x_1 + C_1 \geq 0. \qquad (6)$$

Here the unknown variable is vector $x_1$. The set of equations (6) should not be empty. The sign "$\geq$" in (6) may refer only to certain equations, and in the remaining equations it may be replaced by the sign of strict equality. Further in this problem we shall use the vector $D_1$ – the equation type indicator in (6): if m-equation is an equality, then $D_{1m} = 0$, and in the opposite case $D_{1m} = 1$.

The second problem may be transformed into first problem in the following way. We shall present the restraint (6) in the form

$$T_1^T \cdot x_1 + C_1 - x_2 = 0, \qquad (7)$$

$$x_2 \geq 0. \qquad (8)$$

Let us consider the vector

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad (9)$$

and rewrite the formulas (5, 7, 8) in the form (1, 2, 3) accordingly, where

$$R = \begin{bmatrix} R_1 & Q_{21} \\ Q_{12} & Q \end{bmatrix}, \qquad (10)$$

$$T = \begin{bmatrix} T_1 \\ -E \end{bmatrix}, \qquad (11)$$

$$U = \begin{bmatrix} U_1 \\ V_u \end{bmatrix}, \qquad (12)$$

$$C = C_1, \qquad (13)$$

$$D = \begin{bmatrix} V_d \\ D_1 \end{bmatrix}, \qquad (14)$$

$E$ identity matrix,

$Q$, $Q_{12}$, $Q_{21}$ - zero vectors,

$V_u$, $V_d$ - zero vectors

The M-function for the transformation of the second problem into the first problem has the form:

```
function [R,T,U,C,D]=squ21(R1,T1,U1,C1,D1)
```

### 2.2.4.3. Test for the First Problem

The M-function **function testsqu2()** solves the first problem for a two-dimensional vector **x**. here we are building the graphs shown on Fig. 1, where

- the window `'Error (testsqu2)'` shows the error $\varepsilon$ change depending on iteration number ,
- the window `'Log of error'` shows the error $\ln(\varepsilon)$ change depending on iteration number.
- the window `'Minimum'` shows the function (1) change depending on iteration number.

Let us consider the two-dimensional vector **x** as a point on the plane. The window `'Trajectory'` shows the movement of this point with iteration number growth beginning from the initial point (0, 0) to the final point – the problem's solution.

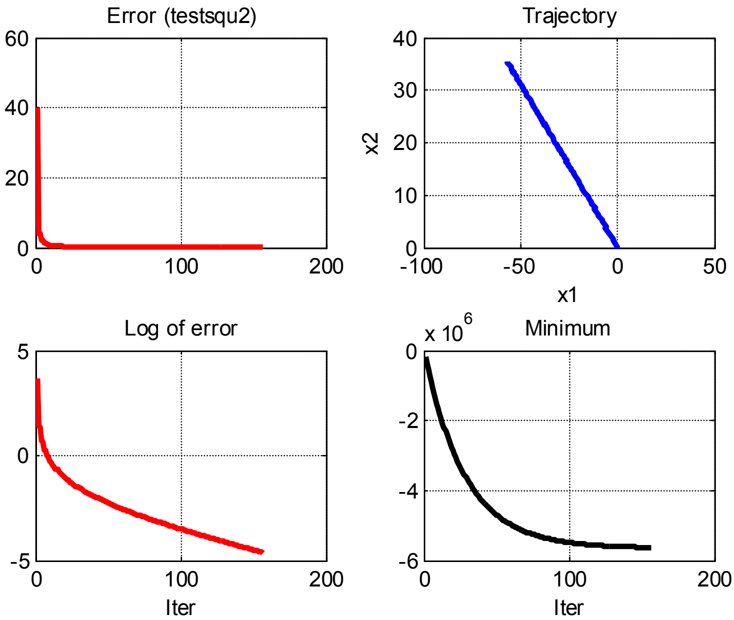



Fig. 1.

#### 2.2.4.4. Test for the Second Problem

The M-function `function testsqu4()` solves the second problem for a four-dimensional vector x. Here the graphs shown on Figure 2 are built, where

- the window `'Error (testsqu2)'` shows the error $\varepsilon$ change depending on iteration number ,
- the window `'Log of error'` shows the error $\ln(\varepsilon)$ change depending on the iteration number,
- the window `'Minimum'` shows the function (1) change depending on iteration number,
- the window `'Trajectory'` shows the change of four components of vector `x` with iteration number growth beginning from the initial value (0) to the final point, that is the problem's solution.

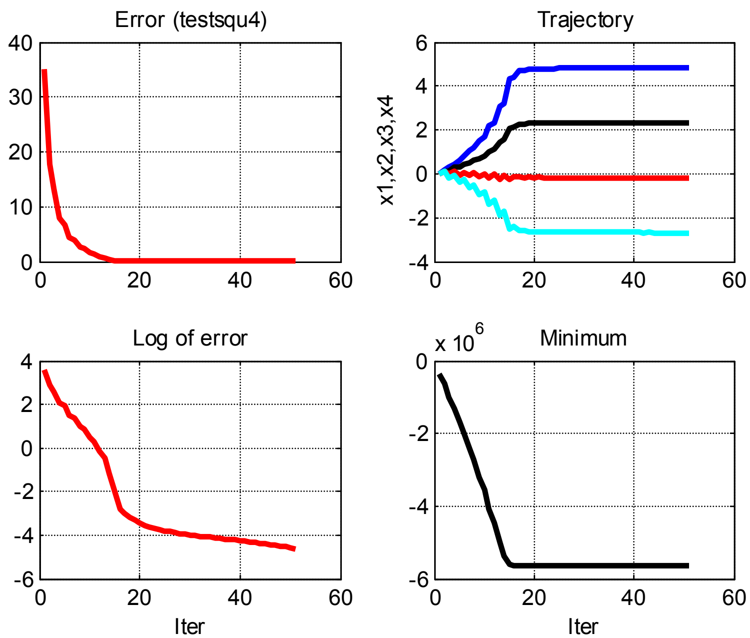



Fig. 2.

# References

1. Dennis J.B. Mathematical Programming and Electrical Networks, New York, 1959, Pages V1, 186 p.
2. Maxwell J.C. Treatise of Electricity and Magnetism, V. 1. M.: Nauka, 1989, p. 328 (in Russian)
3. Khmelnik S.I. Electric Circuits of direct Current for Modelling and Control. Algorithms and Hardware Design. Lulu Inc., ID 293359, ISBN 978-1-4452-0994-42014, Fourth Edition, 2014 (in Russian)

# Programs

## Contents

```
function
[i,erTi,k,n1,mini]=anySLAE2(A,B,r,erd,kmax)
% A*x-B=0
dmin=0.001;
dmax=10000;
    T=A';
    Ct=-B;
    raz=size(T);
for k=[1:1:raz(1)]
    D(k) =0;
    U(k) =0;
    R(k,k) =0;
end
[k,er1K,er2K,erTi,i,ft,n1,yTf,minf,mini]=eqfi(T
,Ct,U,D,r,erd,dmin,dmax,kmax,R);

function  [m_x, m_eps1, m_n1, m_minx,
tmatlab]=anySLAE2m(A,B)
%
tic;
m_x=A\B
sm_x=size(m_x);
tmatlab=toc;
m_n1= A*m_x-B;
sm_n1=size(m_n1);
m_minx=sum(m_n1.^2);
a=sqrt(sum(m_n1.^2)/sm_n1(1));
b=sqrt(sum(m_x.^2)/sm_x(1));
m_eps1=a/b;

function
[k,ft,er2K,n2strogo,n2nestrogo,minf]...
=anySLAE3(A,B,D,r,erd,kmax)
% A*x-B=0
dmin=0.001;
dmax=r;
            U=B;
            T=A;
            rt=size(T);
                for m=[1:1;rt(2)] Ct(m)=0; end
```

```
                    Ct=Ct';
                for m=[1:1;rt(1)]
                    R(m,m)=0;
                end
 [k,er1K,er2K,erTi,i,ft,yt,yTf,minf,mini,p]=...
eqfi(T,Ct,U,D,r,erd,dmin,dmax,kmax,R);
n2=yTf-U;
n2nestrogo=n2.*D';
rD=size(D);
                    for m=[1:1:rD(2)]
                        if D(m)==0
                            DD(m)=1;
                        else
                            DD(m)=0;
                        end
                    end
n2strogo=n2.*DD';

function
[k,er1K,er2K,erTi,i,ft,yTi,yTf,minf,mini,p]...
=eqfi(T,Ct,U,D,r,erd,dmin,dmax,kmax,R)
%
raz=size(T);
for k=[1:1:raz(1)]
    B(k,:) =[ 0,0, R(k,k),  U(k),  D(k)];
end
C=0;
n=0;
[i,f,er1K,er2K,k,p,Un,N,yN,m,ft,yTi,erTi,dD,...
yTf]=rucd2(B,C,r,erd,dmin,dmax,n,T,Ct,kmax);
minf=ft'*ft;
mini=yTi'*yTi;

function figi(kk,erok,Fmink,ii,tit,m)
%
s=size(ii);
subplot(2,2,1); p1=plot(kk,erok,'r');
set(p1,'LineWidth',3)
title(tit);
grid on;
if m==4
```

```
%subplot(2,2,2);
%plot(ii(1,:),ii(2,:),'b',ii(1,:),ii(3,:),'k',
     %ii(1,:),ii(4,:),'r');
    subplot(2,2,2);
    p2=plot(kk,ii(1,:),'b',kk,ii(2,:),'k',...
            kk,ii(3,:),'r',kk,ii(4,:),'c');
    ylabel('x1,x2,x3,x4');
elseif m==2
    subplot(2,2,2);
p2=plot(ii(1,:),ii(2,:),'b');
    ylabel('x2');
    xlabel('x1');
end
set(p2,'LineWidth',3)
title('Trajectory');
grid on;
subplot(2,2,3); p3=plot(kk,log(erok),'r');
set(p3,'LineWidth',3)
title('Log of error');
xlabel('Iter');
grid on;
subplot(2,2,4); p4=plot(kk,Fmink,'k');
set(p4,'LineWidth',3)
title('Minimum');
xlabel('Iter');
grid on;

function [N,nuz] = makingN(branch)
% the creation of the incidence matrix
% b = begN, endN
    raz=size(branch);
    nb=raz(1);
    uz = 0;
    k = 1;
    while k <= nb
        a = branch(k,1);
        if a > uz
            uz = a;
        end
        b = branch(k,2);
        if b > uz
```

```
            uz = b;
        end
        if a > 0
            N(a , k) = -1;
        end
        if b > 0
            N(b , k) = 1;
        end
        k = k + 1;
    end
    nuz=uz;
    n=size(N);
    nuz2=n(1);
    if nuz==nuz2
    else
        nuz
        nuz2
        msg='Numbers of Nodes?'
    end

function [R,U,D] = makingRU(branch)
%  R,U,D from branches
%  b = begN,endN,R,U,D
    raz=size(branch);
    nb=raz(1);
    uz = 0;
    k = 1;
    while k <= nb
        R(k,k) = branch(k,3);
        U(k) = branch(k,4);
        D(k) = branch(k,5);
        k=k+1;
    end
    U=U';

function [i,k,er2K,p,dD]=...
min2_4_3(Rn,Un,erd,D,dmin,dmax,kmax)
%  min function (1.1.24)
i=0*Un;
nUn=sum((Un).^2);
if nUn==0
```

```
    nUn=erd;
end
er2K=999;
raz=size(i);
k=0;
while er2K>erd && k<kmax
    k=k+1;
    [mui1,mui2]=Rdiod(i,D,dmin,dmax);
    p=mui1'+Rn*i-Un;
    np=sum((p).^2);
    er2K=sqrt(np/nUn);
    if er2K==0
        break;
    end
    m2=diag(mui2);
    a=p'*p/(p'*(m2+Rn)*p);
    ap=a*p;
    i=i-ap;
end
dD=mui1;

function
[i,k,p,dD,kk,erok,ero,ogr,Fmin,Fmink,ii]...
= min2_4_300(Rn,Un,erd,D,dmin,dmax,kmax,T,C)
%  min function (1.1.24)
i=0*Un;
nUn=sum(Un.^2);
if nUn==0
    nUn=erd;
end
ero=999;
raz=size(i);
k=0;
while ero>erd && k<kmax
    k=k+1;
    if k==1
        ii=i;
    else
        ii=[ii,i];
    end
    kk(k)=k;
```

```
    [mui1,mui2]=Rdiod(i,D,dmin,dmax);
    p=mui1'+Rn*i-Un;
    m2=diag(mui2);
    a=p'*p/(p'*(m2+Rn)*p);
    ap=a*p;
    i=i-ap;
    for j=[1:1:raz]
        if D(j)==1 && i(j)<0
            i(j)=0;
        end
    end
    ogr=T'*i+C;
    ogr2=ogr'*ogr;
    ei=i'*i;
    ero=sqrt(ogr2/ei);
    erok(k)=ero;
    Fmin=0.5*i'*Rn*i-Un'*i;
    Fmink(k)=Fmin;
end
dD=mui1;

function [f,i,k,er,erk,p,Un]=...
nocondel(R,ro,N,U,C,erd,D,dmin,dmax,kmax)
% basic calculations
Rn=R+ro*N'*N;
Un=U-ro*N'*C;
 [i,k,er,p]=...
    min2_4_3(Rn,Un,erd,D,dmin,dmax,kmax);
f=ro*(N*i+C);
raz=size(N);
nf=sum(f.^2)/raz(1);
ni=sum(i.^2)/raz(2);
erk=sqrt(nf/(ro*ro*ni));

function [f,i,k,er2K,er1K,p,Un,ft,erTi,mui1]...
=nocondel2(R,ro,N,U,C,erd,D,dmin,dmax,...
T,Ct,kmax)
% basic calculations
Rn=R+ro*(N'*N+T*T');
Un=U-ro*(N'*C+T*Ct);
```

```
 [i,k,er2K,p,mui1]=min2_4_3
(Rn,Un,erd,D,dmin,dmax,kmax);
f=ro*(N*i+C);
ft=ro*(T'*i+Ct);
razi=size(i);
razf=size(f);
nf=sum(f.^2)/razf(1);
ni=sum(i.^2)/razi(1);
if ni==0
    er1K=0;
    erTi=0;
else
    er1K=sqrt(nf/(ro*ro*ni));
    razft=size(ft) ;
    nft=sum(ft.^2)/razft(1);
    erTi=sqrt(nft/(ro*ro*ni));
end

function [mui1,mui2]=Rdiod(i,D,dmin,dmax)
% resistance of diodes
    %if D==0
        %mui1=0;mui2=0;
        %return;
    %end
n=size(i);
n=n(1);
k=1;
while k<=n
    if D(k)==1
        if i(k) >0
            mui1(k)=dmin*i(k);
            mui2(k)=dmin;
        else % i(k) =<0
            mui1(k)=dmax*i(k);
            mui2(k)=dmax;
        end
    else
        mui1(k)=dmin*i(k);
        mui2(k)=dmin;
    end
    k=k+1;
```

```
end

function [i,f,er1,er2,k,p,E,N,y,m]
=rucd(b,C,ro,erd,dmin,dmax,n,kmax)
% main function
if sum(C)==0
    m=0;
else
    m=1;
    msg=sum(C)
    i=0;f=0;er1=0;er2=0;k=0;p=0;E=0;N=0;y=0;
    return
end
[N,nuz] = makingN(b);
if nuz>n
    m=2;
    msg=nuz
    i=0;f=0;er1=0;er2=0;k=0;p=0;E=0;N=0;y=0;
    return
end
[R,U,D] = makingRU(b);
[f,i,k,er2,er1,p,E]=nocondel(R,ro,N,U,C,erd,D,d
min,dmax,kmax);
y=f/ro;

function
[i,f,er1K,er2K,k,p,Un,N,yN,m,ft,yTi,erTi,dD,yTf
]=rucd2(b,C,ro,erd,dmin,dmax,n,T,Ct,kmax)
%  main function
if sum(C)==0
    m=0;
else
    m=1;
    msg=sum(C)
    i=0;f=0;er1K=0;er2K=0;k=0;p=0;...
    E=0;N=0;y=0;m=0;ft=0;yt=0;erTi=0;mui1=0;
    return
end
N=0;
[R,U,D] = makingRU(b);
[f,i,k,er2K,er1K,p,Un,ft,erTi,dD]=...
```

```
nocondel2(R,ro,N,U,C,erd,D,dmin,...
dmax,T,Ct,kmax);
yN=f/ro;
yTi=ft/ro;
yTf=T*ft;

function [i,ogr,k,ero,Fmin,kk,erok,Fmink,ii]...
         =squ2(R,T,C,U,D,r,erd,kmax)
%F=0.5*x'*R*x-U'*x --->min, T'*x+C=0, x.*D>=0
dmin=0.001;
dmax=10000;
Rn=R+r*T*T' ;
 Un=U-r*T*C;
 [i,k,p,dD,kk,erok,ero,ogr,Fmin,Fmink,ii]=...
 min2_4_300 (Rn,Un,erd,D,dmin,dmax,kmax,T,C);

function [R,T,U,C,D]=squ21(R1,T1,U1,C1,D1)
%
  Q=0*diag(C1);
  Q21=0*T1;
  Q12=0*T1';
  R=[R1,   Q21;
  Q12, Q];
  U=[U1;0*C1];
  E=diag(0*C1+1);
  T=[T1', -E]';
  D=[0*U1;D1];
  C=C1;

function test()
% B: begN, endN, R, U, D
B(1,:) =[ 2,1, 10,    0,       0];
B(2,:) =[ 2,3, 20,    0 ,      0];
B(3,:) =[ 3,4, 30,    -230,    0];
B(4,:) =[ 4,1, 40,    22 ,     0];
B(5,:) =[ 1,3, 50,    0,       1];
B(6,:) =[ 2,4, 60,    0,       1];
C=1*[0,3,-2,-1]';
dmin=0.001;
dmax=10000;
r=100000;
```

```
erd=0.0001;
n=4;
[i,f,er1,er2,k,p,E,N,y,m]=...
        rucd(B,C,r,erd,dmin,dmax,n);
er1
er2
k
m

function test_anySLAE2()
% A*x-B=0
mode=3;
kmax=10000;
    r=100000;
    erd=0.001;
if mode==1 % 1=ravno
    A=[1,1,9;  2,4,7; 1,0,5];
    B=[1.5,1,3]';
elseif mode==2 % 2=under
    A=[1,1,0,0,1,-1;
          1,0,1,2,0,3];
    B=[1.5,1]';
elseif mode==3 % over
    A=[1,1,0,0,1,-1;
        1,0,1,2,0,3]';
    B=[1.5,1,7,8,9,2]';
elseif mode==4 % 4=over=dispersion
    A=[1,1,1,1,1,1]'
    B=[1,2,3,-4,5,6]';
elseif mode==5 % 5=under=mnogo
    erd=0.0000001;
    m=229;
    T1=[1:1:1+m]';
    T2=[100:1:100+m]';
    A=[T1,T2]';
    B=[1.5,1]';
elseif mode==6 % 6=over=mnogo
    m=125;
    T1=[1:1:1+m];
    T2=[100:1:100+m];
    A=[T1;T2]';
```

```
    B=[1:1:1+m]';
elseif mode==7 % 7ravno, Matrix is singular to
working precision
    A=[1,1,9;  2,2,18; 1,0,5];
    B=[1.5,1,3]';
end
tic
[x,eps1,k,n1,minx]=anySLAE2(A,B,r,erd,kmax)
tmy=toc;
[m_x, m_eps1, m_n1, m_minx,
tmatlab]=anySLAE2m(A,B)

function testqw3()
% B: beg, end, R, U, Diod
kmax=10000;
dmin=0.001;
dmax=1000000;
r=100000;
erd=0.0001;
mode=10;
if mode==1 % 1=ravno
    eq=0;
    A=[1,1,9;  2,4,7; 1,0,5];
    B=[1.5,1,3]';
    D=[0,0,0];
elseif mode==2 % 2=under
    eq=0;
    A=[1,1,0,0,1,-1;
          1,0,1,2,0,3];
    B=[1.5,1]';
    D=[0,0];
elseif mode==3 % over
    eq=0;
    A=[1,1,0,0,1,-1;
          1,0,1,2,0,3]';
    B=[1.5,1,7,8,9,2]';
    D=[0,0,0,0,0,0];
elseif mode==4 % 1=ravno
    eq=1;
    A=[1,1,9;  2,4,7; 1,0,5];
    B=1000*[1,1,1]';
```

```
    D=[0,0,0]; % 0,0,0; 0,1,0; 1,1,0; 1,0,0;
elseif mode==5 % "No active inequalities"
   eq=2;
   A=[1,1,9;  2,4,7; 1,0,5];
   B=1000*[1,1,1]';
   D=[1,1,1]; % 1,1,1; 0,0,1; 0,1,1; 1,0,1;
elseif mode==6 % Bez proverki
   eq=99;
   A=[1,1,9;  2,4,7; 1,0,5];
   B=1000*[1,1,1]';
   D=[0,0,1]; % 1,1,1; 0,0,1; 0,1,1; 1,0,1;
elseif mode==7 % 7=2=under "A must have 2
column(s)" MATLAB trebuet kwadratnuu matrizu
    eq=3;
    A=[1,1,0,0,1,-1;
          1,0,1,2,0,3];
    B=[1.5,1]';
    D=[1,0];
elseif mode==8 %8=3 % over "A must have 6
column(s)" MATLAB trebuet kwadratnuu matrizu
    eq=4;
    A=[1,1,0,0,1,-1;
          1,0,1,2,0,3]';
    B=-[1.5,1,7,8,9,2]';
    D=[1,1,1,1,1,1];
elseif mode==9 %8=3 % over ploho
    eq=99;
    A=[1,1,0,0,1,-1;
          1,0,1,2,0,3]';
    B=[1.5,1,7,8,9,2]';
    D=[0,1,0,1,0,1];
elseif mode==10 %8=3 % over horosho
    eq=99;
    A=[1,1,0,0,1,-1;
          1,0,1,2,0,3]';
    B=[1.5,1,7,8,9,2]';
    D=[1,1,0,1,0,1];
end
[k,ft,er2K,n2strogo,n2nestrogo,minf]=anySLAE3(A
,B,D,r,erd,kmax)
```

```
             if eq==0
                 y=A\B
             elseif eq==1 % D=0
                 Beq=B;
                 Aeq=A;
                 A=[];
                 B=[];
                 x0=[0;0;0];
                 [x,fval] =
fmincon(@myfun2,x0,A,B,Aeq,Beq)
            elseif eq==2 % D=1
                 x0=[0;0;0];
                 %x0=[80;-90;180];
                 [x,fval] =
fmincon(@myfun2,x0,A,B)
           elseif eq==3 % D=1
                 x0=[0;0];
                 [x,fval] =
fmincon(@myfun2,x0,A,B)
           elseif eq==4 % D=1
                 x0=[0;0;0;0;0;0];
                 [x,fval] =
fmincon(@myfun2,x0,A,B)
             end
return
%  X = FMINCON(FUN,X0,A,B,Aeq,Beq) minimizes
FUN subject to the linear
%    equalities Aeq*X = Beq as well as A*X <=
B. (Set A=[] and B=[] if no
%    inequalities exist.)

function testrucd3()
%
R1=11;
R2=12;
R3=13;
U1=100;
U2=0;
U3=0;
D1=0;
D2=1;
```

```
D3=0;
C1=-2;
C2=0;
C3=2;
Ct1=-0.1;
Ct2=0;
Ct3=0;
T=[ 1,0,0;
       2,0,0;
       0,0,0];
B=[1,2,R1,U1,D1;...
   1,3,R2,U2,D2;...
   2,3,R3,U3,D3];
C=[C1,C2,C3]';
Ct=[Ct1,Ct2,Ct3]';
nodes=3;
dmin=0.01;
dmax=1000;
ro=1350;
erd=0.01;
eri=0.01;
kmax=9900;
kimax=8;
io=0;
[i,f,er1,er2,k,p,Un,N,yN,ft,yTi,erTi,dD,yTf,nuz
,m,ki,ro]=...
    rucd3 (B,C,ro,erd,dmin,dmax,nodes,
T,Ct,kmax,eri,kimax,io);
nuz;
m;
i
f
ft
er1_er2=[er1,er2]
k_ki_ro=[k,ki,ro]

function testsqu2()
%
% x=[1,2]'
%  F=0.5*x'*R*x-U'*x --->min,  T'*x+C=0,
x.*D>=0
```

```
dmin=0.001;
dmax=10000;
r=100000;
erd=0.01;
kmax=515;
L=[1,2;
      0,1];
  R=L*L';
  T=[3,5;
     -2,-3]';
  C=-[7,8]';
  U=[1,3]';
  D=[0,1]';
[i,ogr,k,ero,Fmin,kk,erok,Fmink,ii]=squ2(R,T,C,
U,D,r,erd,kmax);
ii
ogr
ero
k
Fmin
figi(kk,erok,Fmink,ii,'Error (testsqu2)',2);

function testsqu4()
%
% x1=[1,2,3,4]'
% x2=[1,2]'
% F=0.5*x1'*R1*x1-U1'*x1 --->min,
% (T1'*x+C1).*D1>=0
dmin=0.001;
dmax=10000;
r=100000;
erd=0.01;
kmax=60000;
L=[1,0,2,0;
   0,5,0,1;
   0,4,0,2;
   0,0,1,1];
  R1=L*L';
  T1=[3,4,5,6;
     -1,-2,-3,-4]';
  C1=-[7,8]';
```

```
  U1=[1,0,3,-1]';
  D1=[1,0]';
  [R,T,U,C,D]=squ21(R1,T1,U1,C1,D1);
%  F=0.5*x'*R*x-U'*x --->min,
%  T'*x+C=0, x.*D>=0
[i,ogr,k,ero,Fmin,kk,erok,Fmink,ii]...
=squ2(R,T,C,U,D,r,erd,kmax);
i
ogr
ero
k
Fmin
figi(kk,erok,Fmink,ii,'Error (testsqu4)',4);
```