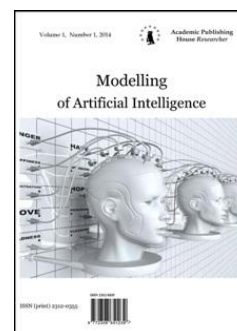


Copyright © 2015 by Academic Publishing House *Researcher*



Published in the Russian Federation  
 Modeling of Artificial Intelligence  
 Has been issued since 2014.  
 ISSN: 2312-0355  
 Vol. 6, Is. 2, pp. 75-81, 2015

DOI: 10.13187/mai.2015.6.75  
[www.ejournal11.com](http://www.ejournal11.com)



UDC 519.853

## Application of Interior Point Method and Newton Method in Large Dimension Flow Distribution Problem

Oleg O. Khamisov

Melentiev Energy Systems Institute SB RAS, Russian Federation  
 Lermontov street 130, Irkutsk city, Irkutsk Region, 664033  
 Laborant  
 E-mail: cygx151@gmail.com

### Abstract

The article is based on sparse matrix techniques applied for solving flow distribution problem. Two methods, applicable for solution of this problem were coded on C++ considering sparse structure of input data.

**Keywords:** flow distribution problem, interior point method, Newton method, sparse matrices.

### Введение

В данной работе рассматривается задача потокораспределения в гидравлических сетях. Математическое описание установившихся режимов движения жидкости в системах труб и методы расчета параметров этих режимов появляется в связи с необходимостью поиска оптимальных режимов функционирования трубопроводных и гидравлических систем. В силу того, что данная задача требует рассмотрения гидравлических цепей большой размерности, возникает необходимость применения методов работы с разреженными структурами данных. Для работы с разреженными матрицами используются алгоритмы символьной факторизации, вложенных сечений и левосторонний метод Холецкого. Приведены результаты численного эксперимента.

### Постановка задачи

Рассматривается задача потокораспределения в следующей постановке [1].

$$F(x) = \sum_{i=1}^n \left( \frac{f_i(x_i)x_i}{1 + \beta_i} - H_i x_i \right) \rightarrow \min, x \in \mathbb{R}^n, \quad (1)$$

$$Ax = Q, A \in \mathbb{R}^{m \times n}, Q \in \mathbb{R}^m, \quad (2)$$

$$x_i \leq x_i, i \in I_1; x_i \leq \bar{x}_i, i \in I_2; \underline{x}_i \leq x_i \leq \bar{x}_i, i \in I_3,$$

где  $f_i(x_i) = s_i |x_i|^{\beta-1} x_i$ ,  $I_1 \subseteq \{1, \dots, n\}$ ,  $I_2 \subseteq \{1, \dots, n\}$ ,  $I_3 \subseteq \{1, \dots, n\}$ . Здесь  $H_i$  – коэффициенты напора,  $s_j$  – коэффициенты сопротивления -й дуги,  $A$  – матрица соединений.  $Q_i$  – значения расхода среды в  $i$ -м узле. Для решения данной задачи используется метод внутренних точек [2].

### Метод внутренних точек

Пусть  $x^0$  – точка, строго удовлетворяющая (2), тогда для  $k \geq 0$ .

$$\sigma_i^k = \begin{cases} (x_i^k - \underline{x}_i)^2, i \in I_1 \\ (x_i^k - \bar{x}_i)^2, i \in I_2 \\ \min \{(x_i^k - \underline{x}_i)^2, (x_i^k - \bar{x}_i)^2\}, i \in I_3 \\ 1, i \notin I_1 \cup I_2 \cup I_3 \end{cases}, \quad (3)$$

$$D^k = \text{diag}(\sigma_1^k, \dots, \sigma_n^k), W^k = AD^k, B^k = W^k A^T, \quad (4)$$

$$c^k = \nabla F(x^k), d^k = W^k c^k, \quad (5)$$

$$u^k: B^k u^k = d^k,$$

$$\delta^k = u^k A - c^k, s^k = D^k \delta^k,$$

$$\lambda_k = \arg \min_{\lambda \in [0,1]} \{F(x^k + \lambda s^k)\},$$

$$x^{k+1} = x^k + \lambda_k s^k,$$

$$\Phi_k = \sum_{i=1}^n \sigma_i \delta_i^2.$$

Условие остановки имеет вид

$$\sqrt{\Phi_k} < \epsilon,$$

где  $\epsilon > 0$  – заданная точность. Если в данной задаче отсутствуют ограничения вида (2), то  $\sigma_i^k = 1 \forall k > 0, i = \overline{1, n}$ , матрицы  $D^k, W^k, B^k$  являются неизменными для любой итерации, и их можно вычислить заранее. Следует отметить, что в силу линейности ограничений (1) и (2) вектор  $x^{k+1}$  всегда является допустимым.

При работе представленной процедуры самыми трудоемкими операциями являются перемножение матриц на шаге (4) и решение системы линейных уравнений на шаге (5). Матрица  $B^k$  является положительно определенной при условии отличия от нуля всех  $x_i$ , поэтому для решения данной системы линейных уравнений используется метод Холецкого. В силу того, что операции на этапах (4) и (5) проводятся над разреженными матрицами большой размерности, основное внимание в данной работе сконцентрировано на применении методов работы с разреженными матрицами для эффективной реализации метода внутренней точки.

### Поиск ненулевой структуры

При проведении операций над разреженными матрицами, во избежание вычисления заведомо нулевых элементов, имеет смысл определить номера строк и столбцов ненулевых элементов (структуру) получаемых матриц или векторов, то есть провести символьную факторизацию. Следует отметить, что структуры матриц  $B^k$ , и матриц Холецкого  $L^k: L^k(L^k)^T = B^k$  остаются неизменными из итерации в итерацию и, как следствие, могут быть найдены до начала итеративного процесса. Далее понадобится следующее определение. Структуры  $i$ -й строки и  $j$ -го столбца матрицы  $A$  определяются следующими формулами:

$$Struct[A_{i*}] = \{j | a_{ij} \neq 0, i > j\}, Struct[A_{*j}] = \{i | a_{ij} \neq 0, i < j\}.$$

Тогда, для матрицы  $B^k = AD^k A^T$  имеет место следующая формула [3]

$$B_{*j} = \bigcup_{i \in A_{j*}} A_{*i},$$

где  $A_{j*} = Struct[A_{j*}], A_{*i} = Struct[A_{*i}], B_{*j} = Struct[B_{*j}] \forall k \geq 0$ . Здесь и далее при вычислении структуры не учитываются численные сокращения.

Для нахождения структуры матрицы Холецкого введем определение графа матрицы. Пусть  $B \in \mathbb{R}^{n \times n}$  – симметричная матрица, неориентированный граф  $G(B) = (V, E)$ , соответствующий данной матрице, определяется следующим образом:  $V = \{1, \dots, n\}, E = \{(i, j) | b_{ij} \neq 0, i < j\}$ . При этом, если  $B$  – ассиметричная матрица, то  $G(B)$  – ориентированный граф с множеством ребер  $E = \{(i \rightarrow j) | b_{ij} \neq 0\}$

Графом заполнения называется неориентированный граф  $G^+(B) = (V^+, E^+)$  симметричной матрицы, содержащий граф  $G(A)$ , а также дополнительные ребра  $(i, j)$  такие, что существует путь в  $G(A)$  из  $i$  в  $j$  через вершины с номерами меньшими, чем  $\min\{i, j\}$ .

Следует отметить, что для любой симметричной положительно-определенной матрицы  $B$  имеет место следующее равенство:

$$G^+(B) = G(L + L^T),$$

где  $L$  – матрица Холецкого. Таким образом, задача поиска структуры матрицы  $L$  сводится к задаче поиска графа  $G^+(B)$ . Нахождение такого графа, согласно его определению, требует выполнения количества операций, соответствующего порядку количества ненулевых элементов матрицы  $B$ . Для улучшения этой скорости используются следующие теоремы и определения.

Деревом исключения матрицы  $B$  называется дерево  $T(B)$ , состоящее из  $n$  вершин. Предок -й вершины определяется следующим образом:

$$parent(i) = \min\{j \mid j > i, (i, j) \in G^+(B)\}.$$

$T(B)$  является деревом поиска в глубину на графе  $G^+(B)$ .

### Поиск символьной структуры треугольной матрицы.

В процессе разложения Холецкого  $LL^T = B$  вычисление -й строки матрицы является решением следующей треугольной системы:

$$L_{0\dots k-1,0\dots k-1}x = B_{0\dots k-1,k},$$

Следует отметить, что множество ненулевых элементов  $B_{0\dots k-1,k}$  совпадает с  $\mathcal{B}_{*k}$ .

Алгоритм для решения такой системы выглядит следующим образом:

```

x = B0...k-1,k
for j = 1 to k - 1 do
  if xj ≠ 0
    for each i: i > j, dij ≠ 0 do
      xi = xi - lijxj

```

В используемой реализации метода Холецкого матрица  $L_{0\dots k-1,0\dots k-1}$  и векторы  $x$  и  $B_{0\dots k-1,k}$  являются разреженными, поэтому появляется необходимость определения ненулевой структуры вектора  $x$ :  $\chi = \{j \mid x_j \neq 0\}$ . Она задается согласно следующим утверждениям:

1.  $b_i \neq 0 \Rightarrow x_i \neq 0$ .
2.  $x_j \neq 0 \wedge \exists i (l_{ij} \neq 0) \Rightarrow x_i \neq 0$ .

Данные утверждения доказывают следующую теорему.

**Теорема 1** [3]. Пусть  $G(L_{0\dots k-1,0\dots k-1})$  – орграф матрицы  $L_{0\dots k-1,0\dots k-1}$ . Тогда структура  $\chi = \{j \mid x_j \neq 0\}$  вектора  $x = L_{0\dots k-1,0\dots k-1}^{-1}p$  определяется следующим образом:

$$\chi = Reach_{L_{0\dots k-1,0\dots k-1}}(\mathcal{B}_{*k}), \quad (7)$$

где  $Reach_{L_{0\dots k-1,0\dots k-1}}(\mathcal{B}_{*k})$  – множество вершин  $d$  таких, что  $\exists (d, p) \in G(L_{0\dots k-1,0\dots k-1}), p \in \mathcal{B}_{*k}$ .

Поиск  $\chi$  может быть осуществлен посредством поиска в глубину на  $G(L_{0\dots k-1,0\dots k-1})$ . Так как  $G^+(B) = G(L + L^T)$ , то  $G(L)$  можно получить из  $G^+(B)$ , добавив ребрам направленность от элементов с меньшими номерами к элементам с большими номерами. Граф  $G(L_{0\dots k-1,0\dots k-1})$  и соответствующее ему дерево исключения  $T(B_{0\dots k-1,0\dots k-1})$ , получаются из  $G(L)$  и  $T(B)$  исключением вершин  $k, \dots, n$  вместе с прилегающими к ним ребрами. В силу того, что  $T(B_{0\dots k-1,0\dots k-1})$  является деревом поиска в глубину для графа  $G(L_{0\dots k-1,0\dots k-1})$ , алгоритм поиска  $\chi$  имеет следующий порядок действий: производится обход дерева  $T(B_{0\dots k-1,0\dots k-1})$  из каждой его вершины, принадлежащей множеству  $\mathcal{B}_{*k}$  по направлению к корню. Тогда множество  $\chi$  будет состоять из всех обойденных вершин.

### Хранение разреженной матрицы

Для описания символьной факторизации матрицы Холецкого необходимо представить способ хранения разреженных матриц на компьютере. Разреженные матрицы хранятся в столбцовом формате. Для хранения произвольной матрицы  $B$  используются три массива, задаваемые следующим образом:

1. Массив  $x$  хранит значения ненулевых элементов матрицы  $B$ , сгруппированные по столбцам.
2. Массив  $e$  хранит столбцовые индексы элементов матрицы  $A$  таким образом, что, если  $x_k$  содержит элемент  $b_{ij}$ , то  $e_k = i$ .

3. Массив  $p$  хранит о первом элементом, на  $-$ м месте хранится увеличенный на единицу индекс последнего элемента массива  $x$ , содержащего элемент  $i$ -го столбца.

### Символьная факторизация для метода Холецкого.

Для получения структуры матрицы  $L$  используется результат следующей теоремы.  
**Теорема 2** [3]. Пусть  $\mathcal{L}_{*i}$  и  $\mathcal{B}_{*i}$  – множества номеров строк ненулевых элементов в  $-$ м столбце матриц  $L$  и  $A$  соответственно (ненулевые шаблоны столбцов). Тогда

$$\mathcal{L}_{*i} = \mathcal{B}_{*i} \cup \{i\} \cup \left( \bigcup_{s \in \{k | i = \text{parent}(k)\}} \mathcal{L}_{*s} \setminus \{s\} \right).$$

Перед непосредственным применением данной теоремы для вычисления номеров ненулевых строк и столбцов элементов матрицы  $L$ , для эффективной работы программы необходимо выделить память под хранение этих номеров, поэтому сначала вычисляется количество ненулевых элементов, и, в силу того, что матрицы хранятся в столбцовом виде, вычисляется количество ненулевых элементов в каждом столбце.

Введем следующие определения. Скелетной матрицей матрицы  $B$  называется  $\hat{B}$  – матрица с наименьшим количеством ненулевых элементов из множества матриц  $\bar{B}$  таких, что  $G^+(\bar{B}) = G^+(B)$ , и задается следующим образом:  $\hat{b}_{ij} \neq 0$  – ненулевой элемент матрицы  $\hat{B}$ , тогда и только тогда, когда  $j$  – лист в  $-$ м строчном поддереве.

Для дерева исключения  $T(B)$   $j$ -м строчным поддеревом называется дерево  $T^j(B)$ , получающееся из дерева путем удаления узлов, не принадлежащих  $\text{Struct}[\mathcal{B}_{*i}]$ , с сохранением наследственных связей между оставшимися элементами.

Наименьший общий предок двух узлов  $a$  и  $b$  некоторого пронумерованного в обратном порядке дерева  $T$  – общий предок  $q$  этих узлов с наименьшим номером и будет далее обозначаться  $q = \text{lca}(a, b)$ .

Вычисление количества ненулевых элементов в столбцах  $c_i = |\mathcal{L}_{*i}|$  может быть выполнено за  $O(|L|)$  операций, где  $|L|$  – количество ненулевых элементов в матрице  $L$ . Из теоремы 2:

$$c_i = |\hat{\mathcal{B}}_{*i}| + \left| \bigcup_{s \in \{k | i = \text{parent}(k)\}} \mathcal{L}_{*s} \setminus \{s\} \right| = |\hat{\mathcal{B}}_{*i}| - e_i + \left| \bigcup_{s \in \{k | i = \text{parent}(k)\}} \mathcal{L}_{*s} \right|,$$

где  $e_i$  – количество потомков  $-$ го узла в дереве исключения,  $\hat{\mathcal{B}}_{*i}$  – структура  $i$ -го столбца скелетной матрицы. Обозначим за  $o_j$  количество повторений номера  $j$  среди структур столбцов, чьи номера являются потомками номера  $i$ , тогда имеет место следующая формула:

$$c_i = |\hat{\mathcal{B}}_{*i}| - e_i + o_i + \sum_{s \in \{k | i = \text{parent}(k)\}} c_s.$$

Отметим, что узел  $j$  принадлежит  $T^i(B)$ , тогда и только тогда, когда  $i \in \mathcal{L}_{*j}$ . Число повторений  $o_j$  может быть определено согласно следующим правилам.

1. Если  $j \notin T^i(B)$ , то  $i \notin \mathcal{L}_{*j}$ , и  $-$ й ряд не влияет на  $o_j$ .
2. Если  $j$  – лист  $T^i(A)$ , то  $\hat{b}_{ij} \neq 0$  в скелетной матрице.  $i$ -я строка не влияет на  $o_j$ , так как  $i$  не является потомком  $j$ .
3. Если  $j$  – не лист  $T^i(B)$ , то обозначим за  $d_{ij}$  количество потомков  $j$  в  $T^i(B)$ .  $i$ -й элемент является ненулевым в каждом из  $d_{ij}$  столбцов, поэтому  $-$ й лист добавляет  $d_{ij} - 1$  повторений к  $o_j$ .

Введем обозначение  $\Delta_i = |\hat{\mathcal{B}}_{*i}| - e_i + o_i$ . Значения  $c_i = \Delta_i + \sum_{s \in \{k | i = \text{parent}(k)\}} c_s$  могут быть найдены при помощи следующего алгоритма:

```
function c_count
     $\Delta_i = 0$  для всех  $i$ 
    for  $k = 1$  to  $n$  do
        if  $k$  – не корень then
             $\Delta_{\text{parent}(i)} = \Delta_{\text{parent}(i)} - 1$ 
        for  $p \in \mathcal{B}_{*k}$  в порядке обратного обхода дерева  $T(B)$  do
```

```

 $\Delta_k = \Delta_k + 1$ 
if  $p$  –не последний элемент в  $\mathcal{B}_{*k}$  then
   $p' =$  элемент в  $\mathcal{B}_{*k}$ , следующий за  $p$ 
   $q = lca(p, p')$ 
   $\Delta_q = \Delta_q - 1$ 
 $c_i = \Delta_i$  для всех  $i$ 
for  $i = 1$  to  $n$  do

```

```

   $c_{parent(i)} = c_{parent(i)} + c_i$ 

```

После нахождения количества ненулевых элементов производится расчет  $\mathcal{L}_{*i}$ . Обозначим через  $\mathcal{L}_{k*} = Struct[L_{k*}]$ . Согласно формуле (7)

$$\mathcal{L}_{k*} = Reach_{L_{0\dots k-1, 0\dots k-1}}(\mathcal{B}_{*k}).$$

Следует отметить, что для вычисления шаблона в формуле не используются числовые значения элементов матрицы, достаточно знать первые  $j - 1$  структур строк. Таким образом, для получения ненулевых шаблонов столбцов матрицы  $L$  используется следующий алгоритм

```

function  $L\_struct$ 
  for  $j = 1$  to  $n$  do
     $\mathcal{L}_{k*} = Reach_{L_{0\dots k-1, 0\dots k-1}}(\mathcal{B}_{*k})$ 
    for  $i \in \mathcal{L}_{k*}$  do
       $\mathcal{L}_{*i} = \mathcal{L}_{*i} \cup \{k\}$ 

```

### Вычисление числовых значений.

Для удобства описания введем две следующие функции:

```

function  $cdiv(k)$ 
   $l_{kk} = \sqrt{b_{kk}}$ 
  for  $i = k + 1$  to  $n$  do
     $l_{ik} = b_{ik} / l_{kk}$ 

```

```

function  $cmod(k, j)$ 
  for  $i = k$  to  $n$  do
     $b_{ik} = b_{ik} - l_{kj} l_{ij}$ 

```

С учетом этих функций, метод Холецкого можно записать в следующем виде

```

for  $k = 1$  to  $n$  do
  for  $j \in \mathcal{L}_{k*}$  do
     $cmod(k, j)$ 
   $cdiv(k)$ 

```

Для описания параллельной версии алгоритма, работающей на многопроцессорных компьютерах с общей памятью отметим приоритет выполнения функций  $cmod$  и  $cdiv$  для  $i$ -й итерации внешнего цикла:

$$cmod(j, i), i = \overline{1, j-1} \rightarrow cdiv(j) \rightarrow cmod(k, j), k = \overline{j+1, n}.$$

В результате можно распараллелить вычисление  $cmod(k, j), k = \overline{j+1, n}$  после  $cdiv(j)$ :

```

for  $k = 1$  to  $n$  do
   $cdiv(k)$ 
  for  $i = k + 1$  to  $n$  do (цикл распараллеливается)
     $cmod(i, k)$ 

```

### Уменьшение заполненности матрицы $L$ .

В работе используется метод внутренних сечений, который путем перестановки строк и столбцов матрицы  $B$  с сохранением симметрии уменьшает заполненность матрицы Холецкого. Идея алгоритма заключается в преобразовании матрицы  $B$  к наиболее близкому блочно-диагональному виду с сохранением симметрии. Таким образом, заполнение при выполнении алгоритма Холецкого будет происходить в рамках таких блоков и практически отсутствовать вне их. Данный алгоритм имеет следующий вид [5]:

1. Положить  $G(B)$  в стек, положить  $N = 1$ .

2. Взять из стека верхний граф  $G = (V, E)$ , если стек пуст, то стоп.
3. Найти псевдо-периферийную вершину  $v$  графа  $G$ .
4. Построить структуру подуровней графа  $G$  из вершины  $v$ , глубина которой пусть будет равна  $M$ .
5. Если  $M < 3$ , то перейти к шагу 6, иначе перейти к шагу 7.
6. Пронумеровать вершины  $G$  числами от  $N$  до  $N + |E|$ , положить  $N \leftarrow N + |E|$ , перейти к шагу 2.
7. Найти минимальный разделитель  $R$  графа  $G$ , принадлежащий подуровню  $\lfloor m/2 \rfloor$ . Пронумеровать вершины  $R$  числами от  $N$  до  $N + |R|$ , положить  $N \leftarrow N + |R|$ .
8. Положить графы, полученные разделением графа  $G$  множеством  $R$  в стек, перейти к шагу 2.

### Использование метода Ньютона

Рассмотрим задачу (1) с ограничениями (2), опустив ограничения (3). Тогда функция Лагранжа для данной задачи будет выглядеть следующим образом:

$$L(x, \lambda) = F(x) + \lambda^T (Ax - Q).$$

В силу выпуклости целевой функции и линейности ограничения условие равенства нулю градиента функции Лагранжа является достаточным условием минимума. На  $(k + 1)$ -м шаге значение  $x^{k+1}$  вычисляется согласно следующей формуле

$$\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} - (\nabla^2 L(x^k, \lambda^k))^{-1} \nabla L(x^k, \lambda^k),$$

откуда  $x^{k+1}$  находится как решение системы

$$\begin{pmatrix} \nabla^2 F(x^k) & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} \nabla^2 F(x^k)x^k - \nabla F(x^k) \\ Q \end{pmatrix}. \quad (8)$$

Здесь  $\nabla^2 F(x)$  – гессиан целевой функции, имеющий вид

$$\nabla^2 F(x) = \text{diag}(s_1 \beta |x_1|^{\beta-1}, \dots, s_n \beta |x_n|^{\beta-1}).$$

Если в некотором  $x^k$  или в решении окажется нули среди компонент вектора  $x_i$ , то гессиан функции Лагранжа окажется вырожденной функцией, и метод Ньютона будет не применим. В таком случае осуществляется переход к методу внутренних точек. В силу неположительной определенности гессиана функции Лагранжа метод Холецкого невозможно использовать для решения системы (8), поэтому используется метод бисопряженных градиентов [6] с учетом симметричности матрицы

### Результаты вычислений

Ниже приводятся результаты предварительного численного эксперимента. Обозначения:  $n$  – количество узлов,  $m$  – количество связей,  $it$  – количество итераций метода внутренних точек,  $T$  – время в секундах. Вычисления производились на компьютере Intel Core i7/2.3GHz/16GB. Программа была написана на C++.

#### Результаты работы программы

$n$	$m$	Вн. Т.		М. Н.		$nz(A)$	$nz(L)$
		$it$	$T$	$it$	$T$		
5000	7078	52	1.64	6	10.1	14153(0.035%)	1744268(7%)
10000	14144	56	11.72	7	28.3	23285(0.016%)	7142814(7.1%)
15000	21200	41	40.21	7	50.1	42397(0.013%)	16416802(7.2%)
20000	28323	55	89.45	7	78.6	56643(0.006%)	28459213(7.1%)
25000	46290	52	583.87	9	121.5	92576(0.008%)	101642294(16%)
30000	55392	58	870.28	9	164.3	110780(0.007%)	132320786(14%)
35000	64474	68	1286.67	9	214.4	128994(0.006%)	170499924(13.9%)
40000	73753	69	2074.64	9	243.6	147502(0.004%)	232206520(14.5%)
50000	92838	61	4116.37	9	351.4	185672(0.004%)	386503091(15.4%)

Тестирование проводилось следующим образом. Сначала задавалось количество узлов  $n$ , затем между узлами случайным образом задавались связи, от 1 до 4 для каждого узла, и подсчитывалось их общее количество. Вектор  $Q$  также задавался случайно, но с условием  $\sum_{i=1}^n Q_i = 0$ . Константы  $s_i$  полагались равными  $10^{-3}$ , константы  $H_i = 0$ .

По результатам тестирования можно сделать следующие выводы. Удалось решить задачи со средними (несколько десятков тысяч переменных, количество переменных здесь равно  $m$ , количество ограничений –  $(n - 1)$ ), согласно терминологии [4], размерностями. Количество итераций практически не зависит от размерности задачи. Наиболее затратная операция – разложение Холецкого (90 % временных затрат), 1 % занимает символьная факторизация, 2 % – вычисление матрицы системы линейных уравнений, остальное – текущие затраты на вычисление скалярных произведений и т.д. на каждой итерации. Решение системы линейных уравнений после разложения Холецкого происходит очень быстро.

В методе Ньютона основное количество вычислений приходится на решение линейной системы уравнений. Согласно результатам вычислений скорость работы бисопряженного алгоритма растет слабее с ростом размерности в сравнении с методом Холецкого.

#### Примечания:

1. Меренков А.П. Теория гидравлических цепей / А.П. Меренков, В.Я. Хасилев. М.: Наука, 1985. 278 с.
2. Дикин И.И. Метод внутренних точек И.И. Дикин. М.: КРАСАНД, 2010. 120 с.
3. Davis T.A. Direct Methods for Sparse Linear Systems / T.A. Davis. SIAM, 2006. 217 p.
4. Нестеров Ю.Е. Введение в выпуклую оптимизацию / Ю.Е. Нестеров. М.: МЦНМО, 2010. – 280 с.
5. Saad Y. Iterative Methods for Sparse Linear Systems (2nd edition) / Y. Saad. SIAM, 1996. 547 p.
6. Pissanetzky S. Sparse Matrix Technology / S. Pissanetzky. Academic Press, 1984.

#### References:

1. Merenkov A.P. Teoriya gidravlicheskih tsepei / A.P. Merenkov, V.Ya. Khasilev. M.: Nauka, 1985. 278 s.
2. Dikin I.I. Metod vnutrennikh tochek I.I. Dikin. M.: KRASAND, 2010. 120 s.
3. Davis T.A. Direct Methods for Sparse Linear Systems / T.A. Davis. SIAM, 2006. 217 p.
4. Nesterov Yu.E. Vvedenie v vypukluyu optimizatsiyu / Yu.E. Nesterov. M.: MTsNMO, 2010. 280 s.
5. Saad Y. Iterative Methods for Sparse Linear Systems (2nd edition) / Y. Saad. SIAM, 1996. 547 p.
6. Pissanetzky S. Sparse Matrix Technology / S. Pissanetzky. Academic Press, 1984.

УДК 519.853

### Применение метода внутренних точек и метода Ньютона к задаче потокораспределения большой размерности

Олег О. Хамисов

Институт систем энергетики имени Л.А. Мелентьева СО РАН, Российская Федерация  
664033, Иркутская область, г. Иркутск, ул. Лермонтова 130  
Лаборант  
E-mail: cygx151@gmail.com

**Аннотация.** Статья основана на использовании техник работы с разреженными матрицами для решения задачи потокораспределения в сети. Проведена на C++ программная реализация двух методов, применяемых для решения данной задачи с учетом разреженной структуры входных данных.

**Ключевые слова:** задача потокораспределения, метод внутренних точек, метод Ньютона, разреженные матрицы.