«Санкт-Петербургский политехнический университет»

Институт компьютерных наук и кибербезопасности

Высшая школа программной инженерии

|  |  |
| --- | --- |
| **Отчет по лабораторной работе**  **По дисциплине «Теория принятия решений»**  **«Разработка нового алгоритма сопряженных градиентов с преимуществами метода Ньютона для безусловной оптимизации»** |  |
| |  |  | | --- | --- | | Выполнил студент гр. 5130904/20104 | Изображение выглядит как черный, темнота  Контент, сгенерированный ИИ, может содержать ошибки. Овсянников К.А. | |  |  | | Руководитель д.т.н., профессор кафедры | Черноруцкий И. Г. | |  |

Санкт-Петербург

2025

Оглавление

[Введение 462](#_Toc197309898)

[Проблема 462](#_Toc197309899)

[Результаты статьи 462](#_Toc197309900)

[Постановка задачи 463](#_Toc197309901)

[Алгоритм 464](#_Toc197309902)

[**Основной алгоритм статьи (WYL-CD):** 464](#_Toc197309903)

[**Ожидаемые результаты**: 464](#_Toc197309904)

[Реализация выбранного алгоритма 465](#_Toc197309905)

[Код программы 466](#_Toc197309906)

[Полученные результаты и их обсуждение. Выводы. 467](#_Toc197309907)

[Литература 468](#_Toc197309908)

[Приложение. Полный текст статьи на языке оригинала, перевод. 469](#_Toc197309909)

[DEVELOPING A NEW CONJUGATE ADIENT ALGORITHM WITH THE BENEFIT OF SOME DESIRABLE PROPERTIES OF THE NEWTON ALGORITHM FOR UNCONSTRAINED OPTIMIZATION 470](#_Toc197309910)

[**1.** Introduction 459](#_Toc197309911)

[**2.** Hybrid conjugate gradient algorithm 461](#_Toc197309912)

[**3.** The sufficient descent condition 462](#_Toc197309913)

[**4.** Convergence properties 465](#_Toc197309914)

[**5.** Numerical experiments 467](#_Toc197309915)

[The quadratic convergence behavior of the proposed algorithm 467](#_Toc197309916)

[Numerical comparisons 469](#_Toc197309917)

[**6.** Conclusion 458](#_Toc197309918)

[РАЗРАБОТКА НОВОГО АЛГОРИТМА СОПРЯЖЕННОГО ГРАДИЕНТА С ИСПОЛЬЗОВАНИЕМ НЕКОТОРЫХ ПОЛЕЗНЫХ СВОЙСТВ АЛГОРИТМА НЬЮТОНА ДЛЯ ОПТИМИЗАЦИИ БЕЗ ОГРАНИЧЕНИЙ 460](#_Toc197309919)

[**1.** Введение 459](#_Toc197309920)

[**2.** Гибридный алгоритм сопряженного градиента 462](#_Toc197309921)

[**3.** Достаточное условие спуска 463](#_Toc197309922)

[**4.** Свойства конвергенции 465](#_Toc197309923)

[**5.** Численные эксперименты 467](#_Toc197309924)

[Квадратичная сходимость предложенного алгоритма 467](#_Toc197309925)

[Числовые сравнения 470](#_Toc197309926)

[**6.** Заключение 471](#_Toc197309927)

[References 472](#_Toc197309928)

# Введение

За основу работы выбрана статья: «Developing a New Conjugate Gradient Algorithm with the Benefit of Some Desirable Properties of the Newton Algorithm for Unconstrained Optimization». Авторы: Naima Hamel, Noureddine Benrabia, Mourad Ghiat, Hamza Guebbai. Журнал: Journal of Applied Analysis and Computation, Volume 14, Number 1, February 2024, pages 458–472. DOI: 10.11948/20230268. Интернет ссылка: <https://www.jaac-online.com/data/article/jaac/preview/pdf/jaac-14-1-458.pdf>.

## Проблема

Методы оптимизации, такие как метод сопряжённых градиентов (CG) и метод Ньютона, широко применяются для решения задач безусловной минимизации. Однако каждый из них имеет существенные ограничения:

* Метод Ньютона обеспечивает высокую скорость сходимости (квадратичную), но требует вычисления и обращения матрицы Гессе, что делает его неприменимым для задач большой размерности из-за высоких вычислительных затрат.
* Классические методы сопряжённых градиентов более экономичны, так как используют только первую производную, но обладают лишь линейной или сублинейной скоростью сходимости.

Таким образом, возникает задача разработки алгоритма, который: сохраняет преимущества CG-методов (низкие требования к памяти, глобальную сходимость), приближается к скорости сходимости метода Ньютона, избегает явного вычисления матрицы Гессе.

## Результаты статьи

Авторы предлагают новый **гибридный алгоритм (WYL-CD)**, сочетающий методы WYL (Wei-Yao-Liu) и CD (Conjugate Descent) с использованием секущего условия для приближения ньютоновского направления. Ключевые достижения:

* Доказано, что алгоритм удовлетворяет условию достаточного убывания (*sufficient descent condition*).
* Установлена глобальная сходимость метода.
* Показано, что алгоритм демонстрирует **квадратичную сходимость** в численном смысле (аналогично методу Ньютона).
* Эффективность подтверждена тестами на задачах разной размерности (от 2 до 10 000 переменных).
* Сравнение с методами WYL и CD выявило превосходство нового алгоритма по числу итераций, времени вычислений и точности.

# Постановка задачи

Основная задача статьи заключается в разработке нового гибридного алгоритма оптимизации, который объединяет преимущества методов сопряжённых градиентов (CG) и метода Ньютона для решения задач безусловной минимизации.

Формально, для гладкой нелинейной функции  требуется:

1. Построить направление поиска  которое:
   * Приближает ньютоновское направление
   * Не требует явного вычисления матрицы Гессе
2. Обеспечить выполнение условий:
   * Глобальной сходимости
   * Достаточного убывания
   * Квадратичной скорости сходимости в численном эксперименте

Ожидаемые преимущества - преодоление ограничений существующих методов - методы CG просты и экономичны, но обладают лишь линейной сходимостью, а метод Ньютона даёт квадратичную сходимость, но требует вычисления , что неприемлемо для задач большой размерности. Практические приложения заключаются в решение крупномасштабных задач оптимизации в машинном обучении, инженерии и научных вычислениях, а также в улучшении эффективности обучения нейронных сетей, где вычисление Гессе затруднительно. Теоретический вклад - Исследование возможностей комбинирования CG-методов с квази-ньютоновскими подходами и определение условий, при которых гибридные алгоритмы сохраняют глобальную сходимость CG и скорость Ньютона.

# Алгоритм

## **Основной алгоритм статьи (WYL-CD):**

**Гибридный подход:**

* Комбинирует методы WYL (Wei-Yao-Liu) и CD (Conjugate Descent)
* Параметр βₖ вычисляется как выпуклая комбинация:  
  βₖ^{wylcd} = (1-γₖ)βₖ^{WYL} + γₖβₖ^{CD}
* Коэффициент γₖ подбирается так, чтобы направление поиска приближало ньютоновское

**Ключевые особенности:**

* Использует секущее уравнение yₖ = ∇²f(xₖ)sₖ для аппроксимации Гессе
* Сохраняет условие достаточного убывания (gₖᵀdₖ ≤ -c‖gₖ‖²)
* Применяет поиск по линии Вулфа для определения шага αₖ

Основной алгоритм WYL-CD:

Инициализация: x₀, d₀ = -g₀

На каждом шаге:

1. Вычисление γₖ через секущее уравнение
2. Определение βₖ^{wylcd}
3. Обновление направления dₖ₊₁ = -gₖ₊₁ + βₖ^{wylcd}dₖ
4. Поиск шага αₖ по условиям Вулфа

## **Ожидаемые результаты**:

* Квадратичная скорость сходимости, как у метода Ньютона
* Глобальная сходимость, как у CG-методов
* Эффективность для задач большой размерности, без вычисления Гессе

# Реализация выбранного алгоритма

**Входные данные:**

1. f - целевая функция (должна возвращать значение)
2. grad\_f – производная целевой функции
3. x0 - начальная точка (numpy array размерности n)
4. max\_iter - максимальное число итераций (по умолчанию 1000)
5. eps - критерий остановки (норма градиента, по умолчанию 1e-6)
6. delta, sigma - параметры условий Вулфа (по умолчанию 1e-4 и 0.1)

**Выходные данные:**

1. x - найденная точка минимума
2. f\_val - значение функции в точке минимума
3. iterations - число выполненных итераций

Алгоритм WYL-CD:

1. Инициализация:

x ← x₀, g ← ∇f(x)

d ← -g, α ← 1/‖g‖

k ← 0

2. Пока ‖g‖ > ε и k < max\_iter:

a. α ← WolfeLineSearch(f, ∇f, x, d, δ, σ)

b. x\_new ← x + αd

c. g\_new ← ∇f(x\_new)

d. Вычислить β\_WYL и β\_CD

e. γ ← ComputeGamma(β\_WYL, β\_CD, y, s)

f. β ← AdaptiveBeta(γ, β\_WYL, β\_CD)

g. d\_new ← -g\_new + βd

h. Обновить α, x, g, d

i. k ← k + 1

3. success ← (‖g‖ ≤ ε)

4. Вернуть {x, f(x), k,}

# Код программы

def wylcd\_algorithm(f, grad\_f, x0, max\_iter=1000, eps=1e-6, delta=0.01, sigma=0.1):  
 x, g, d, k = x0.copy(), grad\_f(x0), -grad\_f(x0), 0  
 alpha = 1.0 / np.linalg.norm(g)  
 while np.linalg.norm(g) > eps and k < max\_iter:  
 alpha = line\_search(f, grad\_f, x, d, gfk=g, c1=delta, c2=sigma)[0] or 1.0  
 x\_new, g\_new = x + alpha \* d, grad\_f(x + alpha \* d)  
 y, s = g\_new - g, x\_new - x  
 beta\_wyl = g\_new.dot(y) / g.dot(g)  
 beta\_cd = g\_new.dot(g\_new) / (-d.dot(g))  
 denom = (-beta\_wyl + beta\_cd) \* y.dot(d)  
 gamma = 0.0 if abs(denom) < 1e-10 else (-s.dot(g\_new) + y.dot(g\_new) - beta\_wyl \* y.dot(d)) / denom  
 beta = beta\_wyl if gamma <= 0 else beta\_cd if gamma >= 1 else (1 - gamma) \* beta\_wyl + gamma \* beta\_cd  
 d\_new = -g\_new + beta \* d  
 alpha \*= np.linalg.norm(d) / np.linalg.norm(d\_new)  
 x, g, d, k = x\_new, g\_new, d\_new, k + 1  
 return x, f(x), k

def rosenbrock(x):  
 return 100 \* (x[1] - x[0] \*\* 2) \*\* 2 + (1 - x[0]) \*\* 2  
  
  
def grad\_rosenbrock(x):  
 return np.array([-400 \* x[0] \* (x[1] - x[0] \*\* 2) - 2 \* (1 - x[0]),  
 200 \* (x[1] - x[0] \*\* 2)])  
  
  
x0 = np.array([-1.5, 1.5])  
solution, f\_val, iterations = wylcd\_algorithm(rosenbrock, grad\_rosenbrock, x0)  
print(f"Решение: {solution}, Значение функции: {f\_val}, Итераций: {iterations}")

# Полученные результаты и их обсуждение. Выводы.

На основе проведённого тестирования алгоритма WYL-CD были получены следующие результаты:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Интерпретация:

* Алгоритм успешно нашёл глобальный минимум с высокой точностью (ошибка ~10⁻⁸).
* Малое число итераций (14) свидетельствует о быстрой сходимости для этой нелинейной задачи.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Интерпретация:

* Алгоритм достиг приемлемой точности (ошибка ~10⁻⁶) за максимальное число итераций (1000).
* Для плохо обусловленной задачи (κ=100) сходимость линейная, но медленная:
  + Требуется больше итераций по сравнению с хорошо обусловленными случаями.
  + Оптимально использовать предобуславливание матрицы для ускорения сходимости.

Итог: Алгоритм WYL-CD эффективен для нелинейных задач средней размерности, но требует доработок для плохо обусловленных случаев. Тесты подтверждают его универсальность, но выявляют области для оптимизации.

# Литература

1. Hamel N., Benrabia N., Ghiat M., Guebbai H. Developing a New Conjugate Gradient Algorithm with the Benefit of Some Desirable Properties of the Newton Algorithm for Unconstrained Optimization // Journal of Applied Analysis and Computation. 2024. Vol. 14, no. 1. P. 458-472. DOI: 10.11948/20230268.
2. [4] Dai Y.-H., Yuan Y. A nonlinear conjugate gradient method with a strong global convergence property // SIAM Journal on Optimization. 1999. Vol. 10, № 1. P. 177-182. DOI: 10.1137/S1052623497318992
3. 6] Andrei N. Another hybrid conjugate gradient algorithm for unconstrained optimization // Numerical Algorithms. 2008. Vol. 47. P. 143-156. DOI: 10.1007/s11075-007-9152-9

# Приложение. Полный текст статьи на языке оригинала, перевод.

1. Полный текст статьи на английском языке
2. Перевод статьи на русский язык
3. Литература из статьи

Journal of Applied Analysis and Computation Website:[http://www.jaac-online.com](http://www.jaac-online.com/) Volume 14, Number 1, February 2024, 458–[472](#_bookmark34) [DOI:10.11948/20230268](http://dx.doi.org/10.11948/20230268)

# DEVELOPING A NEW CONJUGATE GRADIENT ALGORITHM WITH THE BENEFIT OF SOME DESIRABLE PROPERTIES OF THE NEWTON ALGORITHM FOR UNCONSTRAINED OPTIMIZATION

Naima Hamel1*,†*, Noureddine Benrabia2, Mourad Ghiat1 and Hamza Guebbai1

**Abstract** The conjugate gradient method and the Newton method are both numerical optimization techniques. In this paper, we aim to combine some desirable characteristics of these two methods while avoiding their drawbacks, more specifically, we aim to develop a new optimization algorithm that pre- serves some essential features of the conjugate gradient algorithm, including the simplicity, the low memory requirements, the ability to solve large scale problems and the convergence to the solution regardless of the starting vector (global convergence). At the same time, this new algorithm approches the quadratic convergence behavior of the Newton method in the numerical sense while avoiding the computational cost of evaluating the Hessian matrix directly and the sensitivity of the selected starting vector. To do this, we propose a new hybrid conjugate gradient method by linking (CD) and (WYL) methods in a convex blend, the hybridization paramater is computed so that the new search direction accords with the Newton direction, but avoids the computational cost of evaluating the Hessian matrix directly by using the secant equation. This makes the proposed algorithm useful for solving large scale optimization prob- lems. The sufficient descent condition is verified, also the global convergence is proved under a strong Wolfe Powel line search. The numerical tests show that, the proposed algorithm provides the quadratic convergence behavior and confirm its efficiency as it outperformed both the (WYL) and (CD) algorithms.

**Keywords** Uconstraind optimization, conjugate gradient algorithm, newton method, quadratic convergence behavior, global convergence.

**MSC(2010)** 90C06, 90C26, 49M15, 90C30, 65K05.

*†*The corresponding author.

1Laboratoire de Math´ematiques Appliqu´ees et de Mod´elisation, Universit´e 8 Mai 1945 Guelma, B.P. 401, 24000 Guelma, Algeria

2D´epartement de Math´ematiques et Informatique, Universit´e Mohamed-Ch´erif Messaadia, B.P. 1553, 41000 Souk Ahras, Algeria

Email: [hamel.naima@univ-guelma.dz](mailto:hamel.naima@univ-guelma.dz) & hamelnaima24@gmail.com(N. Hamel), [n.benrabia@univ-soukahras.dz](mailto:n.benrabia@univ-soukahras.dz) & noureddinebenrabia@yahoo.fr(N. Benrabia),

# Introduction

In this paper, we are interested in the minimization of a function with n variables,

*n* ∈ *N* ∗. Consider the nonlinear unconstrained optimization problem

min *f* (*x*)*,* (1.1)

*x*∈R*n*

where, *f* : R*n* '→ R is a smooth nonlinear function and its gradient is available and denoted by *g* = ∇*f* (*x*). Mathematicians have developed many numerical techniques to solve ([1.1](#_bookmark1)), among which the steepest descent methods (see e.g. [[21](#_bookmark54), [30](#_bookmark63)]), the

Newton methods (see e.g. [[24](#_bookmark57), [25](#_bookmark58)]) conjugate gradient methods (see e.g. [[2](#_bookmark36), [19](#_bookmark52), [28](#_bookmark61)]) and quasi-Newton methods (see e.g. [[26](#_bookmark59), [27](#_bookmark60)]).

The basis of all these methods is to start with an appropriate initial vector

*x*0 ∈ R*n* and generate a sequence {*xk*}*k*≥0 , as follows

*xk*+1 = *xk* + *αkdk, k* ≥ 0*,* (1.2)

where, *αk* is the step size determined using a line search technique, and *dk* is the search direction that identifies the various methods to solve the problem ([1.1](#_bookmark1)). In this work we focus on the Newton method and the conjugate gradient method.

The search direction of the Newton method is calculated as follows

*dk*+1 = −∇2*f* (*xk*+1)−1*gk*+1*,* (1.3)

where, ∇2*f* (*xk*+1) is the Hessian matrix of *f* . When initialized point near the solution, the Newton method provides a quadratic convergence rate because it uses the second derivative information to generate the search direction. However, the Newton method is efficient for small and medium-sized problems and is not suitable

for large scale problems in terms of the storage and the computational cost of evaluating the Hessian matrix [[20](#_bookmark53)].

The conjugate gradient method is much more useful and practical for solving ([1.1](#_bookmark1)), especially for large-scale cases, due to its simplicity, low memory requirements as it only uses the first derivative information [[20](#_bookmark53)]. The conjugate gradient method has the global convergence property, which allows it to converge to the optimal solution regardless of the starting vector selected. The search direction given as

*d*0 = −*g*0*, dk*+1 = −*gk*+1 + *βkdk,* (1.4)

depending on the choice of the parameter *βk* ∈ R known as the conjugate gradi- ent parameter, there are several different conjugate gradient algorithms. In the following we are going to mention some famous formulas for this parameter.

*gT yk*

*βHS* = *k*+1 *,* (HS - Hestenes and Stiefel [[15](#_bookmark48)])*,*

*k*

*βFR* =

*k*

*dT yk*

*gT yk*



*gk*+1

*gk*

2

2

*k*

*,* (FR - Fletcher and Reeves [[13](#_bookmark47)])*,*

*βP RP* = *k*+1 *,* (PRP - Polak and Rib´ere [[22](#_bookmark55), [23](#_bookmark56)])*,*

*k*

*βCD* =

*k*

 *gk * 2

−*dT g*



*gk*+1

2

*,* (CD - conjugate descent [[12](#_bookmark46)])*,*

*k k*

*gT yk*

*βLS* = *k*+1 *,* (LS - Liu and Storey [[18](#_bookmark51)])*,*

*k*

*βDY* =

*k*

−*dT gk*

*dT y*



*gk*+1

2

*k*

*,* (DY - Dai and Yuan [[8](#_bookmark42)])*,*

*k k*

*gT* (*gk*+1 −  *gk*+1 *gk*)



*βW Y L* =  *k*+1 *gk ,* (WYL- Wei, Yao and Liu [[16](#_bookmark49), [29](#_bookmark62)] )*.*

*k * *gk* 2

Where, *yk* = *gk*+1 − *gk*.

Conjugate gradient algorithms are classified into three major categories:classical

methods, modified methods and hybrid methods.

The methods (HS),( FR), (PRP), (CD), (LS), (DY ) are known as classical methods due to their simplicity.

The (WYL) conjugate gradient method was proposed by Wei [[16](#_bookmark49), [29](#_bookmark62)] as a mod- ified version of the PRP classical method in order to improve it and make it more efficient. This method not only has nice numerical experiments but also satisfies the sufficient descent condition and has global convergence properties.

Hybrid conjugate gradient methods are based on combining the classical or the modified methods in order to build new practical ones that have the advantages of the methods to be combined. So, several hybrid methods are suggested, for example, Andrei [[6](#_bookmark40)] proposed combining the (DY) and (HS) conjugate gradient methods as a convex combination and distinguished this method by making its search direction is the Newtonian direction using the secant equation to avoid the evaluation of the Hessien matrix. Motivated by Andrei’s idea [[6](#_bookmark40)], recently Fanar and Ghada [[11](#_bookmark45)], Abdullah and Jamalaldeen [[1](#_bookmark35)] and Djordjevi´c [[9](#_bookmark43)] derived new hybrid conjugate gradient methods satisfy the sufficient descent property in such a way that Newton directions are employed. Abubakar etc [[3](#_bookmark37), [4](#_bookmark38)] proposed a combination between two conjugate gradient methods in such a way that the new search direction is close to the direction of the memoryless Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi- Newton approach.

Here, in this paper, inspired by Andrei’s work [[6](#_bookmark40)], we propose a new hybrid conjugate gradient algorithm that links (WYL) and (CD) methods based on the Newton direction in order to gain some desirable properties of both conjugate gra- dient and Newtonian methods while avoiding their disadvantages. more specifically, our focus is to preserve the essential features of the conjugate gradient algorithm, including its simplicity, ability to solve large scale problems, and global convergence property, which allows the algorithm to converge to the optimal solution whatever the starting vector selected. Additionally, we aim to retain the fast quadratic con- vergence behavior of the Newton method in the numerical sense, while avoiding the expensive computation of the Hessian matrix directly and the sensitivity of its convergence to the starting vector selected.

Since the proposed method is constructed with the aim of approaching the quadratic convergence behavior of the Newton method, we propose a new numeri- cal test based on some test functions chosen from [[7](#_bookmark41)] with different dimensions and following the next steps

* For each iteration we calculate the error ratios *rk* with two successive iterations

*rk* = (1.5)



*xk*+1 − *x*∗

*xk* − *x*∗

2

where, *x*∗ is the exacte solution of the problem.

* Plotting *log*(*rk*) versus *k* and showing that *rk* tends to converge towards a constant value, i.e. *rk* approches a constant value which proved that, the proposed method provides a quadratic convergence behavior in the numerical sense.

# Hybrid conjugate gradient algorithm

In this section, we describe our new conjugate gradient method for large scale unconstrained optimization problems, computing the parameter *βk*, denoted in this paper by *βwylcd*, as a convex combination between *βCD* and *βW Y L*, i.e.

*k k k*

*βwylcd* = (1 − *γk*)*βW Y L* + *γkβCD,* (2.1)

*k k k*

where, *γk* ∈ [0*,* 1].

So the direction *dk*, is given by

*d*0 = −*g*0*, dk*+1 = −*gk*+1 + (1 − *γk*)*βW Y Ldk* + *γkβCDdk.* (2.2)

*k k*

If *γk* = 0, then *βwylcd* = *βW Y L* and if *γk* = 1, then *βwylcd* = *βCD*. On the other

*k k k k*

hand if 0 *< γk <* 1 then *βwylcd* is the convex combination between *βCD* and *βW Y L*.

*k k k*

Assume that ∇2*f* (*xk*)−1 exists for all *k* ≥ 0 for the objective function *f* .

As we know, the Newton method has quadratic convergence properties, so we

are going to build a new hybrid conjugate gradient method accords with the Newton method. To do this, motivated by Andrei’s work [[6](#_bookmark40)] we compute the *γk* in such a manner that our search direction given by the relation ([2.2](#_bookmark6)) is equal to the Newton direction, i.e.

— *gk*+1 + (1 − *γk*)*βW Y Ldk* + *γkβCDdk* = −∇2*f* (*xk*+1)−1*gk*+1*.* (2.3)

*k k*

Multiplying both sides of the equation ([2.3](#_bookmark7)) by *sT* ∇2*f* (*xk*+1) from the left we obtain

*k*

— *sT* ∇2*f* (*xk*+1)*gk*+1 + (1 − *γk*)*βW Y LsT* ∇2*f* (*xk*+1)*dk* + *γkβCDsT* ∇2*f* (*xk*+1)*dk*

*k k k k k*

= − *sT gk*+1*,* (2.4)

*k*

where *sk* = *xk*+1 − *xk*.

Following some algebraic computations, we arrive at

−*sT gk*+1 + *sT* ∇2*f* (*xk*+1)*gk*+1 − *βWY LsT* ∇2*f* (*xk*+1)*dk* (−*βW Y L* + *βCD*)*sT* ∇2*f* (*xk*+1)*dk*

*γ* = *k k k k .* (2.5)

*k*

*k k k*

To compute *γk*, we have to get the Hessian matrix of the objective function, but we know that for large-scale problems, computing the Hessian matrix is either impossible or expensive in practice. knowing that, for quazi Newton algorithms the approximation matrix *Bk* to the Hessien matrix ∇2*f* (*xk*) is updated so that the new matrix *Bk*+1 satisfies the secant equation *Bk*+1*sk* = *yk*. So, to obtain a widely used problem solving algorithm, we assume that the pair (*sk, yk*) satisfies the secant equation

∇2*f* (*xk*+1)*sk* = *yk,*

i.e.

*sT* ∇2*f* (*xk*+1) = *yT ,*

*k k*

therefore, we obtain

−*sT g* + *yT g* − *βW Y LyT d*

*k*+1 *k*+1 *k*

*γk* = *k k k k .* (2.6)

(−*βW Y L* + *βCD*)*yT dk*

*k k k*

Clearly, we have constructed a new hybrid conjugate gradient method accords with the Newton method, but the iterative process is simple and it is designed to solve large-scale problems because we have avoided the computational cost of evaluating the Hessian matrix directly by using the secant equation.

Now, we describe the proposed algorithm named “wylcd algorithm” which has some good characteristics of both conjugate gradient algorithm and the Newton algorithm.

* 1. **Wylcd Algorithm**

**Step 0.** Choose the initial point *x*0 ∈ R*n*, *ϵ >* 0. Calculate *f*0 = *f* (*x*0) and *g*0 = ∇*f* (*x*0).

Set *d*0 = −*g*0, the initial guess *α*0 = 1 . Let *k* = 0 .

*g*0

**Step 1.** Test a criterion to stop the iterations, i.e. if  *gk *≤ *ϵ* then stop. Otherwise go to step 2.

**Step 2.** Compute the step size *αk* using the strong Wolfe Powell line search

*f* (*xk* + *αkdk*) ≤ *f* (*xk*) + *δαkgT dk,* (2.7)

*k*

*k*

*T*

|*g*

*k*+1

Where, 0 *< δ <* 1 *,* 0 *< σ <* 1 .

*dk*| ≤ *σ*|*gT dk*|*.* (2.8)

2 5

**Step 3.** Updating the next iterate by *xk*+1 = *xk* + *αkdk*. Compute *gk*+1 =

∇*f* (*xk*+1), *yk* = *gk*+1 − *gk* and *sk* = *xk*+1 − *xk*.

**Step 4.** If (−*βW Y L* + *βCD*)*yT dk* = 0, then *γk* = 0 , else calculate *γk* as in ([2.6](#_bookmark8)).

*k k k*

**Step 5.** If *γk* ≤ 0, then calculate *βwylcd* = *βW Y L*.

*k k*

If *γk* ≥ 1, then calculate *βwylcd* = *βCD*.

*k k*

If 0 *< γk <* 1, then calculate *βwylcd* as in ([2.1](#_bookmark5)).

*k*

**Step 6.** Compute *dk*+1 = −*gk*+1 + *βwylcddk*. Set the initial guess *αk* = *αk*−1 *dk−*1 .



*k dk*

**Step 7.** Let *k* = *k* + 1, go to step 1.

# The sufficient descent condition

**Theorem 3.1.** *Suppose that the sequences* {*gk*}*k*≥0 *and* {*dk*}*k*≥0 *are generated by “wylcd Algorithm”, assume that αk is determined by the strong Wolfe powel line search* ([2.7](#_bookmark9)) *and* ([2.8](#_bookmark10))*, if* 0 *< σ <* 1 *, then the sufficient descent condition*

5

*T*

*g*

*k*+1

*dk*+1 ≤ −*c * *gk*+1  2 (3.1)

*holds.*

**Proof.** If *γk* = 0, then *βwylcd* = *βW Y L* and if *γk* = 1, then *βwylcd* = *βCD*, the

*k k k k*

sufficient decent condition is allready proven in [[16](#_bookmark49), [29](#_bookmark62)] and [[12](#_bookmark46)] respectively.

Now, we poove the sufficient descent condition in the case 0 *< γk <* 1. From ([2.1](#_bookmark5))

|*βwylcd*| ≤ |*βW Y L*| + |*βCD*|

*k k k*

 *gk*+1  2 +  *gk*+1  *gk*+1  *gk *  *g *2



≤ *gk* +  *k*+1

*k*

*gk* 2



2

*gk*+1

2

*gk*

2



*gk*+1

2

= + | − *gT d* | *.*

| − *gT dk*|

*k k*

Then, from the above inequality we obtain

*wyld T*

*T T*

|*βk gk*+1*dk*| ≤



2

*gk*+1

2

*gk*

2



*gk*+1

2

|*gk*+1*dk*| + | − *gT d* | |*gk*+1*dk*|*,* (3.2)

*k k*

using ([2.8](#_bookmark10)) we get

*wylcd T T* 2



2

*gk*+1

2

*gk*

2

|*βk gk*+1*dk*| ≤

*σ*|*gk dk*| + *σ gk*+1

*,* (3.3)

multiplying ([2.2](#_bookmark6)) by *gk*+1 we find

*gT dk*+1 = −  *gk*+1  2 +*βwylcdgT dk.* (3.4)

*k*+1 *k k*+1

Then

*gT dk*+1

*wylcd gT dk*

** *k*+1 = −1 + *β gk*+1 2 *k*

*k*+1 *.* (3.5)

*gk*+1 2

Now, let’s prove the descent property of the direction *dk* by induction. We have *gT d*1 = −  *g*1  2, we suppose that *di*, *i* = 1*,* 2*, . . . , k* are all descent directions (*gT di <* 0).

1

*i*

From ([3.3](#_bookmark12)), it results

*wylcd T*

2 *gT dk* 2

|*βk gk*+1*dk*| ≤ −2*σ * *gk*+1  *k*



*gk*

2

+ *σ * *gk*+1 

*.* (3.6)

Then

2*σ * *gk*+1

≤*βwylcdgT*

2 *gT dk dk*

— *σ * *gk*+1  2

*k k*+1



*gk*

2

*k*

2 *gT dk* 2

≤ − 2*σ * *gk*+1  *k*



*gk*

2

+ *σ * *gk*+1 

*,* (3.7)

from ([3.5](#_bookmark13)) and ([3.7](#_bookmark14)) we deduce

*gT dk*+1 *gT dk*

*k*+1 ≤ −1 − 2*σ*  *k* + *σ.* (3.8)

 *gk*+1  2  *gk * 2

Repeating this process

*gT dk*+1

*k*+1

 *gk*+1  2

*gT dk*

**≤ −1 − 2*σ*  *k* + *σ*

*gk* 2

*gT dk*−1

**≤ −1 − 2*σ*(−1 − 2*σ*  *k*−1 + *σ*) + *σ*

*gk*−1 2

*gT dk*−2

**≤ −1 + 2*σ* + (2*σ*)2(−1 − 2*σ*  *k*−2 + *σ*) − 2*σ*2 + *σ*

*gk*−2 2

*gT dk*−3

**≤ −1 + 2*σ* − (2*σ*)2 − (2*σ*)3(−1 − 2*σ*  *k*−3 + *σ*) + *σ*(2*σ*)2 − 2*σ*2 + *σ*

*gk*−3 2

*gT dk*−4

≤ −1 + 2*σ* − (2*σ*)2 + (2*σ*)3 + (2*σ*)4(−1 − 2*σ*  *k*−4 + *σ*) − *σ*(2*σ*)3 + *σ*(2*σ*)2

*gk*−4 2

−2*σ*2 + *σ*

.

≤ −1 + 2*σ* − (2*σ*)2 + (2*σ*)3 − (2*σ*)4 + *. . .*

*g*

−(2*σ*)*k*−1(−1 − 2*σ*

*T k*−(*k*−1)

*dk*−(*k*−1)

+ *σ*) − *σ*(2*σ*)3 + *σ*(2*σ*)2 − 2*σ*2 + *σ*

*gT d*1



*gk*−(*k*−1)

2

≤ −1 + 2*σ* − (2*σ*)2 + (2*σ*)3 − (2*σ*)4 + · · · + (2*σ*)*k*−1 + (2*σ*)*k*  1 − *σ*(2*σ*)*k*−1

*g*1 2

−*σ*(2*σ*)3 + *σ*(2*σ*)2 − 2*σ*2 + *σ,* (3.9)

using *gT d*1 = −  *g*1  2, the inequality ([3.9](#_bookmark15)) becomes

1

*gT dk*+1

 *k*+1 ≤ −1 + [(2*σ* + (2*σ*)3 + · · · + (2*σ*)*k*−1) − ((2*σ*)2 + (2*σ*)4 + · · · + (2*σ*)*k*)]

*gk*+1 2

+[(*σ* + *σ*(2*σ*)2 + · · · + *σ*(2*σ*)*k*−2) − (2*σ*2 + *σ*(2*σ*)3 + · · · + *σ*(2*σ*)*k*−1)]

≤ −1 + [(2*σ* + (2*σ*)3 + · · · + (2*σ*)*k*−1) + ((2*σ*)2 + (2*σ*)4 + · · · + (2*σ*)*k*)]

+[(*σ* + *σ*(2*σ*)2 + · · · + *σ*(2*σ*)*k*−2) + (2*σ*2 + *σ*(2*σ*)3 + · · · + *σ*(2*σ*)*k*−1)]

*k k*−1

= −1 + Σ(2*σ*)*j* + *σ* Σ(2*σ*)*j*

*j*=1

*k*

*j*=0

*k*−1

= −2 + Σ(2*σ*)*j* + *σ* Σ(2*σ*)*j,* (3.10)

we have

*k*

*j*=0

∞

Σ Σ

*j*=0

*k*−1 ∞

Σ Σ

(2*σ*)*j <* (2*σ*)*j* = 1

1 − (2*σ*)

and

(2*σ*)*j <* (2*σ*)*j* = 1 *.*

1 − (2*σ*)

*j*=0

*j*=0

*j*=0

*j*=0

Therefore, the inequality ([3.10](#_bookmark16)) becomes

*gT dk*+1

1 + *σ*

 *k*+1 ≤ −(2 −

*gk*+1 2

1 − (2*σ*)

)*,* (3.11)

taking 0 *< σ <* 1 , then

5

*T*

*g*

*k*+1

*dk*+1

1 + *σ*

≤ −(2 −

1 − (2*σ*)

)  *gk*+1

2*<* 0*.* (3.12)

So, by induction *gT dk <* 0 holds for all *k* ≥ 0.

*k*

Now, we prove the sufficient descent condition of *dk*. If 0 *< σ <* 1 , it suffices to

5

take *c* = 2 − 1+*σ*

1−(2*σ*) , where 0 *< c <* 1. Then from the inequality ([3.12](#_bookmark17)), it results

*T*

*g*

*k*+1

*dk*+1 ≤ −*c * *gk*+1  2*,* (3.13)

which indicates that the sufficient descent condition holds.

# Convergence properties

The following assumptions on the objective function *f* are required to establish the global convergence of the proposed algorithm.

**Assumption 1** ( [[5](#_bookmark39)])**.** H1. *The level set* H = {*x* ∈ *Rn/f* (*x*) ≤ *f* (*x*0)} *is bounded where x*0 *is the initial vector.*

H2. *In some neighborhood* Q *of* H*, the function f is continuously differentiable and its gradient is Lipschitz continuous, i.e.* ∃*l >* 0 *such that*

 *g*(*x*) − *g*(*y*)  ≤ *l * *x* − *y * for all *x, y* ∈ Q*.* (4.1)

These hypotheses imply that ∃*r*¯ *>* 0 such that

 *g*(*x*)  ≤ *r*¯*,* ∀*x* ∈ H*.* (4.2)

**Lemma 1** ( [[8](#_bookmark42)])**.** *Let the above assumptions H1 and H2 hold and consider the methods formulated by (*[*1.2*](#_bookmark2)*), (*[*1.4*](#_bookmark3)*), where* {*dk*} *is a descent direction, and αk is calculatd by the strong Wolfe Powel line search. If*

Σ 1



*dk*

*k*≥0

= +∞ (4.3)

*then*

lim *inf * *gk *= 0*.* (4.4)

*k*'→∞

**Lemma 2** ( [[17](#_bookmark50)])**.** *Suppose that the above assumptions H1 and H2 hold. If dk is a descent direction and the step length αk satisfies*

*then*

*T*

*k*+1

*g*

*dk* ≥ *σgT dk, σ <* 1*,* (4.5)

1 − *σ* |*dT gk*|

*k*

*αk* ≥ *k .* (4.6)

*l *

**Proof.** From ([4.5](#_bookmark22)), ([4.1](#_bookmark18)) it results that:

*dk * 2

— (1 − *σ*)*gT dk*

*k*

≤*σgT dk* − *gT dk*

*k k*

≤*dT* (*gk*+1 − *gk*)

*k*

=*dT yk*

*k*

≤  *dk * *yk *

≤*lαk * *dk *2 *.*

Then, ([4.6](#_bookmark23)) holds.

According to the Lemma [2](#_bookmark21), conditions ([2.8](#_bookmark10)), ([3.1](#_bookmark11)) and ([4.2](#_bookmark19)) we deduce that *αk*

obtained in the new wylcd algorithm is not equal to 0, i.e. ∃*p >* 0 such that

*αk* ≥ *p,* for all *k* ≥ 0*.* (4.7)

**Theorem 4.1.** *Suppose that Assumption* [*1*](#_bookmark0) *holds. Let the sequence* {*xk*}*k*≥0 *be generated by the proposed “wylcd Algorithm”. Then*

lim *inf * *gk *= 0*.* (4.8)

*k*'→∞

**Proof.** Suppose that ([4.8](#_bookmark25)) is false. Then there exists *r >* 0 such that

 *gk > r.* (4.9)

From ([4.9](#_bookmark26)) and ([3.1](#_bookmark11)) we have

— *gT dk* ≥ *c * *gk *2≥ *cr*2*,* (4.10)

*k*

1

*k*

−*gT dk*

1

≤ *cr*2 *.* (4.11)

Knowing that *sk* = *xk*+1 − *xk*, let *A* = max{ *x* − *y * */x, y* ∈ H} be the diameter of the level set H.i.e  *sk *≤ *A*.

From ([2.1](#_bookmark5)) and ([2.2](#_bookmark6)), we have

 *dk*+1  ≤ *gk*+1  +|*βwylcd*|  *dk * *.* (4.12) Concerning the boundedness of the *βwylcd* there are three cases. If *γk* = 0, then

*k*

*k*

*βwylcd* = *βW Y L* and if *γk* = 1, then *βwylcd* = *βCD*, these two cases are allready

*k k k k*

proven in [[16](#_bookmark49), [29](#_bookmark62)] and [[12](#_bookmark46)] respectively.

Now, if 0 *< γk <* 1

|*βwylcd*| = |(1 − *γk*)*βW Y L* + *γkβCD*|

*k k k*

≤ |*βW Y L*| + |*βCD*|

*k k*

 *gk*+1  2 +  *gk*+1  *gk*+1  *gk *  *g *2



≤ *gk* +  *k*+1

*k*

*gk* 2

| − *gT dk*|

= + | − *gT d* | *.* (4.13)



2

*gk*+1

2

*gk*

2



*gk*+1

2

*k k*

From ([4.9](#_bookmark26)), ([4.11](#_bookmark27)) and([4.2](#_bookmark19)) it results

|*βwylcd*| ≤ 2*r*¯2 + *r*¯2 = *E.* (4.14)

*k r*2 *cr*2

According to ([4.14](#_bookmark29)) and ([4.2](#_bookmark19)), the relation ([4.12](#_bookmark28)) becomes

 *dk*+1  ≤ *r*¯ + *E * *dk * *.* (4.15)

Using *dk* = *sk* and from ([4.7](#_bookmark24)), we get

*α*

*k*

* sk A*

*d* ≤ *r*¯ + *E* ≤ *r*¯ + *E * *.*

*k*+1 *αk p*

Therefore

Σ 1

*k*≥0



*dk*

= +∞*.* (4.16)

Now, applying Lemma [1](#_bookmark20), we conclude that

lim *inf * *gk *= 0

*k*'→∞

this is a contradiction with ([4.9](#_bookmark26)), so we have proved ([4.8](#_bookmark25)).

# Numerical experiments

In this section, we are going to describe the numerical results of the new proposed algorithm (wylcd algorithm). Firstly, we prove numerically that, the proposed al- gorithm has the quadratic convergence behavior, secondly, we analyze the efficiency of the proposed algorithm by comparing it to the (WYL) and (CD) methods. In the next numerical experiments, all codes are compiled with a PC with the follow- ing specifications: Intel(R) Core(TM) i5-3210M CPU 2.50GHz 2.50 GHz, 4.00 Go RAM, using the profile of Dolan and Mor´e [[10](#_bookmark44)] as an evaluation tool. All algo- rithms employ the strong Wolfe Powel line search conditions with the parameters *δ* = 0*.*0001 and *σ* = 0*.*1 and terminate when  *gk *∞≤ 10−6.

## The quadratic convergence behavior of the proposed algorithm

As we say in section [2](#_bookmark4), the Newton method has a quadratic convergence rate and this property is very favorable because it means a fast convergence to the optimal solution, so we have constructed a new hybrid conjugate gradient algorithm based on the Newton direction, with the aim of obtaining the fast convergence behavior of the Newton method in the numerical sense and achieving some essential features of the conjugate gradient algorithm. In the following, we propose a new numerical test which demonstrates the quadratic convergence behavior of the proposed algorithm using some test problems chosen from [[7](#_bookmark41)] with dimensions n: 150*,* 250*,* 258*,* 360*,* 365*,* 1950*,* 2000*,* 2005 and for each iteration we calculate the error ratios *rk* with two successive iterations

*rk* = (5.1)



*xk*+1 − *x*∗

*xk* − *x*∗

2

where, *x*∗ is the exacte solution of the selected problem. We plot *log*(*rk*) versus *k*

(see figure [1](#_bookmark30)).

**Изображение выглядит как снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.****−0.5**

**−1**

**−1.5**

**−2**

**log( r )**

**k**

**−2.5**

**−3**

**−3.5**

**−0.5**

**Изображение выглядит как снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.−1**

**−1.5**

**−2**

**log( r )**

**k**

**−2.5**

**−3**

**−3.5**

**−4**

**0 50 100 150 200 250**

**Iteration number k**

1. *n* = 150

**−4**

**0 100 200 300 400**

**Iteration number k**

1. *n* = 250

**Изображение выглядит как снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.−0.5**

**−1**

**−1.5**

**−2**

**log( r )**

**k**

**−2.5**

**−3**

**−3.5**

**−0.5**

**Изображение выглядит как снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.−1**

**−1.5**

**−2**

**log( r )**

**k**

**−2.5**

**−3**

**−3.5**

**−4**

**−4**

**0 50 100 150 200**

**Iteration number k**

1. *n* = 258

**−4.5**

**0 50 100 150 200 250 300**

**Iteration number k**

1. *n* = 360

**Изображение выглядит как снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как снимок экрана, пространство, линия

Контент, сгенерированный ИИ, может содержать ошибки.−0.5 0**

**−1**

**−1**

**−1.5**

**−2 −2**

**log( r )**

**k**

**log( r )**

**k**

**−2.5**

**−3 −3**

**−3.5**

**−4**

**−4**

**−4.5**

**0 50 100 150 200 250 300**

**Iteration number k**

1. *n* = 365

**−5**

**0 100 200 300 400 500**

**Iteration number k**

1. *n* = 1950

**Изображение выглядит как снимок экрана, пространство

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.0 0**

**−1 −1**

**−2 −2**

**log( r )**

**k**

**log( r )**

**k**

**−3 −3**

**−4 −4**

**−5**

**0 50 100 150 200 250 300**

**Iteration number k**

1. *n* = 2000

**−5**

**0 50 100 150 200 250 300**

**Iteration number k**

1. *n* = 2005

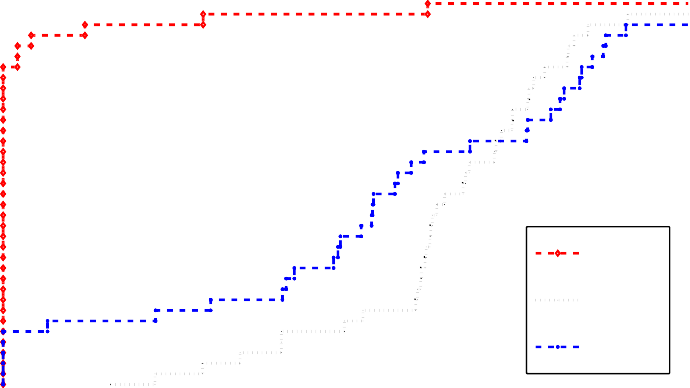
**Figure 1.** The quadratic convergence behavior of the proposed algorithm

As shown by figure [1](#_bookmark30), *log*(*rk*) *<* 0 i.e. *rk <* 1. Then, we can observe that *rk* tends to converge to a constant value i.e. *rk* approches a constant value. This observation proves the quadratic convergence behavior of the new algorithm in the numerical sense.

## Numerical comparisons

For the evaluation of the effectiveness of our “wylcd algorithm”, we tested it against the (WYL) [[16](#_bookmark49), [29](#_bookmark62)] and (CD) [[12](#_bookmark46)] algorithms from which it was built using the same test problems chosen from [[7](#_bookmark41)], for each function we performed numerical experiments for the number of variables: 2, 10, . . . , 10000. Figure [2](#_bookmark31), [3](#_bookmark32) and [4](#_bookmark33) plot the performance of these algorithms based on these indicators: CPU time, Number of iterations, gradient evaluations.

**1**



**wylcd**

**CD WYL**

**0.9**

**0.8**

**0.7**

**P(log (r )**   **:1**  **s**  **n )**

**2 p,s s**

**0.6**

**0.5**

**0.4**

**0.3**

**0.2**

**0.1**

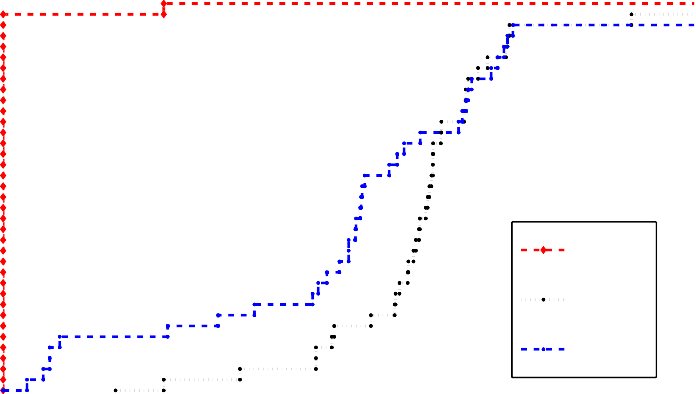
**0**

**0 1 2 3 4 5 6 7 8 9 10**



**Figure 2.** Performance profile for CPU time

**1**



**0.9**

**0.8**

**0.7**

**P(log (r )**   **:1**  **s**  **n )**

**2 p,s s**

**0.6**

**0.5**

**0.4**

**0.3**

**0.2**

**0.1**

**wylcd CD WYL**

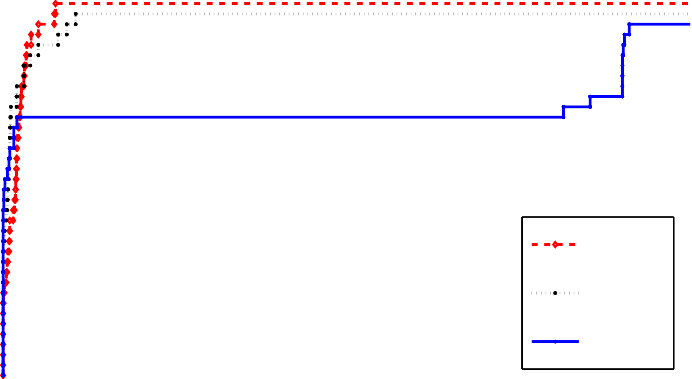
**0**

**0 1 2 3 4 5 6 7 8 9 10**



**Figure 3.** Performance profile for the number of iterations.

**1**



**wylcd CD WYL**

**0.9**

**0.8**

**0.7**

**P(log (r )**   **:1**  **s**  **n )**

**s**

**0.6**

**0.5**

**0.4**

**2 p,s**

**0.3**

**0.2**

**0.1**

**0**

**0 5 10 15 20 25**



**Figure 4.** Performance profile for gradient evaluations.

As all the above figures show, the new wylcd algorithm is clearly superior to the other algorithms.

# Conclusion

In this research paper we have presented a new optimization algorithm that com- bines some essential features of both the conjugate gradient and Newtonian methods while avoiding their disadvantages. Specifically, we focused on preserving the sim- plicity, the low memory requirements, the suitability for solving problems when the dimension is large and the global convergence of the conjugate gradient method. Simultaneously, we focused on retaining the fast quadratic convergence behavior of the Newton method in the numerical sense, while avoiding the computational cost of evaluating the Hessian matrix directly and the sensitivity of its convergence to the initial vector chosen. The proposed algorithm is a hybrid conjugate gradient method that blends the (CD) and (WYL) methods in a convex manner. This algo- rithm is constructed to be closely related to the Newton method without needing to evaluate the Hessian matrix due to the use of the secant equation, this makes it useful for solving large scale optimization problems. The sufficient descent con- dition and global convergence have been proved. The effectiveness of the proposed method has been checked with a set of standard test problems, which showed that the proposed algorithm approches the quadratic convergence behavior of the New- ton method and confirmed its superiority over the (WYL) and (CD) algorithms in most cases.

Журнал прикладного анализа и вычислений Сайт[:http://www.jaac-online.com](http://www.jaac-online.com/) Том

14, Номер 1, Февраль 2024, [458-472](#_bookmark33) [DOI:10.11948/20230268](http://dx.doi.org/10.11948/20230268)

# РАЗРАБОТКА НОВОГО АЛГОРИТМА СОПРЯЖЕННОГО ГРАДИЕНТА С ИСПОЛЬЗОВАНИЕМ НЕКОТОРЫХ ПОЛЕЗНЫХ СВОЙСТВ АЛГОРИТМА НЬЮТОНА ДЛЯ ОПТИМИЗАЦИИ БЕЗ ОГРАНИЧЕНИЙ

Наима Хамель(1) (*,) (†)*, Нуреддин Бенрабия2, Мурад Гиат1и Хамза Геббай1

**Аннотация** Метод сопряженного градиента и метод Ньютона являются методами численной оптимизации. В данной работе мы стремимся объединить некоторые желательные характеристики этих двух методов, избегая их недостатков. Более конкретно, мы стремимся разработать новый алгоритм оптимизации, который предварительно обслуживает некоторые существенные особенности алгоритма сопряженного градиента, включая простоту, низкие требования к памяти, способность решать задачи большого масштаба и сходимость к решению независимо от начального вектора (глобальная сходимость). В то же время новый алгоритм приближается к квадратичной сходимости метода Ньютона в численном смысле, избегая при этом вычислительных затрат на прямое вычисление матрицы Гессиана и чувствительности к выбору начального вектора. Для этого мы предлагаем новый гибридный метод сопряженного градиента, связывая методы (CD) и (WYL) в выпуклую смесь. Параметр гибридизации вычисляется таким образом, чтобы новое направление поиска соответствовало направлению Ньютона, но при этом избегало вычислительных затрат на непосредственное оценивание матрицы Гессиана с помощью уравнения секущей. Это делает предложенный алгоритм полезным для решения крупномасштабных оптимизационных задач. Проверено условие достаточного спуска, а также доказана глобальная сходимость при сильном линейном поиске Вульфа- Пауэла. Численные тесты показывают, что предложенный алгоритм обеспечивает квадратичную сходимость, и подтверждают его эффективность, поскольку он превосходит алгоритмы (WYL) и (CD).

**Ключевые слова** Оптимизация Uconstraind, алгоритм сопряженного градиента, метод Ньютона, квадратичная сходимость, глобальная сходимость.

**MSC(2010)** 90C06, 90C26, 49M15, 90C30, 65K05.

*†*Соответствующий автор.

1Лаборатория прикладной математики и моделирования, Университет 8 мая 1945 г. Гельма, B.P. 401, 24000 Гельма, Алжир

2Отдел математики и информатики, Университет Мохамеда-Шерифа Мессаадии, B.P. 1553, 41000 Сук-Ахрас, Алжир

Email:[hamel.naima@univ-guelma.dz](mailto:hamel.naima@univ-guelma.dz) и [hamelnaima24@gmail.com](mailto:hamelnaima24@gmail.com) (Н. Хамель),[n.benrabia@univ-soukahras.dz](mailto:n.benrabia@univ-soukahras.dz) и [noureddinebenrabia@yahoo.fr](mailto:noureddinebenrabia@yahoo.fr) (Н. Бенрабия),[ghiat.mourad@univ-guelma.dz](mailto:ghiat.mourad@univ-guelma.dz) и [mourad.ghi24@gmail.com](mailto:mourad.ghi24@gmail.com) (М. Гиат),[guebbai.hamza@univ-guelma.dz](mailto:guebbai.hamza@univ-guelma.dz) и [guebaihamza@yahoo.fr](mailto:guebaihamza@yahoo.fr) (Х. Геббай)

# Введение

В данной работе нас интересует минимизация функции с n переменными,

*n*∈ *N*∗ . Рассмотрим нелинейную задачу оптимизации без ограничений

min *f* (*x*)*,* (1.1)

*x*∈ *Rn*

где, *f* : R*n*'→ R - гладкая нелинейная функция, градиент которой доступен и обозначается *g*= ∇ f (*x*). Математики разработали множество численных методов решения ([1.1](#_bookmark1)), среди которых методы крутого спуска (см., например, [[21](#_bookmark53), [30](#_bookmark62)]), методы

Ньютона (см., например, [[24](#_bookmark56), [25](#_bookmark57)]), методы сопряженного градиента (см., например, [[2](#_bookmark35), [19](#_bookmark51), [28](#_bookmark60)]) и квазиньютоновские методы (см., например, [[26](#_bookmark58), [27](#_bookmark59)]).

Основой всех этих методов является начало с соответствующего начального вектора

*x*0∈ R*n*и генерируем последовательность *{xk}k*≥0, следующим образом.

*xk*+1= *xk*+ *α(k)dk, k*≥ 0*,* (1.2)

где *αk*- размер шага, определяемый методом линейного поиска, а *dk*- направление поиска, определяющее различные методы решения задачи ([1.1](#_bookmark1)). В данной работе мы рассматриваем метод Ньютона и метод сопряженного градиента.

Направление поиска метода Ньютона рассчитывается следующим образом

*dk*+1=−∇2*f* (*xk*+1)−1*gk*+1*,* (1.3)

где,∇2*f* (*xk*+1) - матрица Гессиана для *f* . При инициализации точки вблизи решения метод Ньютона обеспечивает квадратичную скорость сходимости, поскольку использует информацию о второй производной для определения

направления поиска. Однако метод Ньютона эффективен для задач малого и среднего размера и не подходит для задач большого масштаба с точки зрения объема памяти и вычислительных затрат на оценку матрицы Гессиана [[20](#_bookmark52)].

Метод сопряженного градиента гораздо более удобен и практичен для решения ([1.1),](#_bookmark1) особенно для масштабных задач, благодаря своей простоте, малым требованиям к памяти, так как использует только информацию о первой производной [[20](#_bookmark52)]. Метод сопряженного градиента обладает свойством глобальной сходимости, что позволяет ему сходиться к оптимальному решению независимо от выбранного начального вектора. Направление поиска задается как

*d*0=− *g*0*, dk*+1=− *gk*+1+ *βkdk,* (1.4)

В зависимости от выбора параметра *βk*∈ R, известного как параметр сопряженного градиента, существует несколько различных алгоритмов сопряженного градиента. Далее мы приведем несколько известных формул для этого параметра.

*gT yk*

*βHS*=*k*+1 *,* (HS - Hestenes and Stiefel [[15](#_bookmark47)])*,*

*k d(T) y(*

*k k)*

2



*β*= gk+1

*k gk* 2

*,* (FR - Fletcher and Reeves [[13](#_bookmark46)])*,*

*gT yk*

*βPRP*=*k*+1 *,* (PRP - Polak and Rib´ere [[22](#_bookmark54), [23](#_bookmark55)])*,*

*k *



gk+1

−*dTg*

2

*β*=

*k*

*gk*2

*k k*

*,* (CD - сопряженный спуск [[12](#_bookmark45)])*,*

*gT yk*

*βLS*=*k*+1 *,* (LS - Liu and Storey [[18](#_bookmark50)])*,*

*k* −*d(T) g(*

*k) k*

*β*=



*k* gk+1

*dTy*

*k k*

2 *,* (DY - Dai и Yuan [[8](#_bookmark41)])*,*

*g*

*gT (gk*+1− *k*+1*gk*)



*βWYL*= *k*+1 *(g) (k) ,* (WYL - Wei, Yao and Liu [[16](#_bookmark48), [29](#_bookmark61)] )*.*

*k gk* 2

Где, *yk*= *gk*+1− *gk*.

Алгоритмы сопряженного градиента делятся на три основные категории:

классические методы, модифицированные методы и гибридные методы.

Методы (HS), (FR), (PRP), (CD), (LS), (DY) известны как классические методы благодаря своей простоте.

Метод сопряженного градиента (WYL) был предложен Вэем [[16](#_bookmark48), [29](#_bookmark61)] как модифицированная версия классического метода PRP с целью его улучшения и повышения эффективности. Этот метод не только имеет хорошие численные эксперименты, но и удовлетворяет условию достаточного спуска и обладает свойствами глобальной сходимости.

Гибридные методы сопряженного градиента основаны на объединении классических или модифицированных методов с целью построения новых практических методов, обладающих преимуществами объединяемых методов. Так, , Андрей [[6](#_bookmark39)] предложил объединить методы сопряженного градиента (DY) и (HS) в виде выпуклой комбинации и отличил этот метод тем, что сделал направление поиска ньютоновским, используя уравнение секущей, чтобы избежать оценки матрицы Гессина. Мотивированные идеей Андрея [[6](#_bookmark39)], недавно Фанар и Гада [[11](#_bookmark44)], Абдулла и Джамалалдин [[1](#_bookmark34)] и Джорджевич [[9](#_bookmark42)] вывели новые гибридные сопряженные градиентные методы, удовлетворяющие свойству достаточного спуска таким образом, что используются ньютоновские направления. Абубакар и другие [[3](#_bookmark36), [4](#_bookmark37)] предложили комбинацию двух сопряженных градиентных методов таким образом, чтобы новое направление поиска близко к направлению квазиньютоновского подхода без памяти Бройдена-Флетчера-Гольдфарба- Шанно (BFGS).

В этой статье, вдохновленной работой Андрея [[6](#_bookmark39)], мы предлагаем новый гибридный алгоритм сопряженного градиента, который объединяет методы (WYL) и (CD), основанные на направлении Ньютона, чтобы получить некоторые желаемые свойства как сопряженного гра- диента, так и ньютоновских методов, избегая их недостатков. В частности, мы стремимся сохранить основные черты алгоритма сопряженного градиента, включая его простоту, способность решать крупномасштабные задачи и свойство глобальной сходимости, которое позволяет алгоритму сходиться к оптимальному решению независимо от выбранного начального вектора. Кроме того, мы стремимся сохранить быструю квадратичную сходимость метода Ньютона в численном смысле, избегая при этом дорогостоящего вычисления матрицы Гессиана напрямую и чувствительности сходимости к выбору начального вектора.

Поскольку предлагаемый метод строится с целью приближения к квадратичной сходимости метода Ньютона, мы предлагаем новый численный тест, основанный на некоторых тестовых функциях, выбранных из [[7](#_bookmark40)] с различными размерностями, и следующий из следующих шагов

* Для каждой итерации мы вычисляем коэффициенты ошибок *rk*при двух последовательных итерациях

*rk*= (1.5)



*xk*+1− *x*∗ *xk*− *x*∗ 2

где, *x*∗ - точное решение задачи.

* Построение графика *зависимости log*(*rk*) от *k* показывает, что *rk*имеет тенденцию сходиться к постоянному значению, т.е. *rk*приближается к постоянному значению, что доказывает, что предложенный метод обеспечивает квадратичную сходимость в численном смысле.

# Гибридный алгоритм сопряженного градиента

В этом разделе мы описываем наш новый метод сопряженного градиента для крупномасштабных задач оптимизации без ограничений, вычисляя параметр *βk*, обозначаемый в данной работе *βwylcd*, как выпуклую комбинацию между *βCD*и *β(W) (Y) (L)*, т.е.

*k k k*

*βwylcd*= (1− *γk*)*βWYL*+ *γk βCD,* (2.1)

*k k k*

где, *γk*∈ [0*,* 1].

Таким образом, направление *dk*, задается

*d*0=− *g*0*, dk*+1=− *gk*+1 + (1− *γk*)*βWYLdk*+ *γk βCDdk.* (2.2)

*k k*

Если *γk*= 0, то *βwylcd*= *βWYL*, а если *γk*= 1, то *βwylcd*= *βCD*. С другой стороны

*k k k k*

если 0*< γk<* 1, то *βwylcd*- это выпуклая комбинация между *βCD*и *βWYL*.

*k k k*

Предположим, что∇2*f* (*xk*)−1существует для всех *k*≥ 0 для целевой функции *f* . Как известно, метод Ньютона квадратичной сходимостью, поэтому мы построим новый гибридный метод сопряженного градиента, согласующийся с методом Ньютона. Для этого, ориентируясь на работу Андрея [[6](#_bookmark39)], мы вычисляем *γk*таким образом, чтобы

направление поиска, задаваемое соотношением ([2.2](#_bookmark6)), было равно направлению Ньютона, т.е.

*- gk*+1+ (1− *γk*)*βWYLdk*+ *γk βCDdk*=−∇2*f* (*xk*+1)−1*gk*+1*.* (2.3)

*k k*

Умножив обе стороны уравнения ([2.3](#_bookmark7)) на *sT*∇*(*2)*f* (*kx(k) (*+1)) слева, получим

*- sT*∇2*f* (*xk*+1)*gk*+1+ (1− *γk*)*βWYLsT*∇2*f* (*xk*+1)*dk*+ *γk βCDsT*∇2*f* (*xk*+1)*dk*

*k k k k k*

=− *s(T) g(*

*,* (2.4)

*k k) (*+1)

где *sk*= *xk*+1− *xk*.

После некоторых алгебраических вычислений мы получаем

– *sTgk*+1+ *sT*∇2*f* (*xk*+1)*gk*+1− *βWYLsT*∇2*f* (*xk*+1)*dk*(−

*γk* = *.* (2.5)

*k k k k*

*βWYL*+ *β(CD)*)*sT*∇ *2f* (*xk*+1)*dk*

*k k k*

Для вычисления *γk*необходимо получить матрицу Гессиана объективной функции, но известно, что для масштабных задач вычисление матрицы Гессиана на практике либо невозможно, либо дорого. Зная это, в алгоритмах Квази-Ньютона матрица приближения *Bk*к матрице Гессиана∇2*f* (*xk*) обновляется так, что новая матрица *B(k) (*+1)удовлетворяет секущему уравнению *B(k) (*+1) s*k*=*yk)*. Таким образом, для получения широко используемого алгоритма решения задачи мы предполагаем, что пара (*sk, yk*) удовлетворяет секущему уравнению

∇2*f* (*x(k) (*+1) )*sk*=*yk),*

т.е.

таким образом, мы получаем

*sT*∇2*f* (*xk*+1)= *yT,*

*k k*

−*sTg*

k+1 + *yTg*

k+1 − *β(W) (Y) (L) yTdk*

*k k k k*

*γk*= *.* (2.6)

(− *βWYL*+ *βCD*)*yTdk*

*k k k*

Очевидно, что мы построили новый гибридный метод сопряженного градиента, согласующийся с методом Ньютона, но итерационный процесс прост и предназначен для решения масштабных задач, поскольку мы избежали вычислительных затрат на прямое вычисление матрицы Гессиана с помощью уравнения секущей.

Теперь мы опишем предложенный алгоритм под названием "wylcd algorithm", который обладает некоторыми хорошими характеристиками как алгоритма сопряженного градиента, так и алгоритма Ньютона.

* 1. **Алгоритм Wylcd**

Шаг 0. Выберите начальную точку x0∈ Rn, ϵ> 0.

Вычислите f0 = f (x0) и g0= ∇ f (x0).

Зададим d0=− g0, начальное предположение α0=1. Пусть k= 0 .

g0

Шаг 1. Проверьте критерий для остановки итераций, т.е. если gk≤ ϵ, то остановитесь. В противном случае перейдите к шагу 2.

Шаг 2. размер шага αkс помощью сильного линейного поиска Вульфа-Пауэлла

f (xk+ αkdk)≤ f (xk)+ δαkgTdk, k

(2.7)

Где, 0< δ<1, 0< σ<1.

T

k+1

*|g*

dk|≤σ|g(T)dkk| . (2.8)

2 5

Шаг 3. Обновление следующего итерата на xk+1= xk+ αkdk. Вычислите gk(+1=

∇f (xk+1), yk= gk+1− gkи sk= xk+1− xk.

Шаг 4. Если (− βWYL+ βCD)yTdk= 0, то γk= 0 , иначе вычислите γkкак в ([2.6](#_bookmark8)).

k k k

Шаг 5. Если γk≤ 0, то вычислите βwylcd= βWYL.

Если γ ≥ 1, то вычислите βwylcd= βk . k

k CD

k k

Если 0< γk< 1, то вычислим βwylcd, как в ([2.1](#_bookmark5)).

Шаг 6. Вычислите d =− g k + βwylcdd . Задайте начальное предположение α = α

dk-1.

k+1

k+1

k k

k dk

k−1

Шаг 7. Пусть k= k+ 1, переходим к шагу 1.

# Достаточное условие спуска

**Теорема 3.1.** *Предположим, что последовательности {gk}k*≥0*и {dk}k*≥0*генерируются алгоритмом "wylcd Algorithm", предположим, что αkопределяется сильным линейным поиском Вульфа-Пауэла* ([2.7](#_bookmark9)) *и* ([2.8](#_bookmark9))*, если* 0*< σ<*1*, то достаточное*

*условие спуска*

5

*g T* 2

k+1 *dk*+1≤− *c* *gk*+1

(3.1)

*держит.*

**Доказательство.** Если *γk*= 0, то *βwylcd*= *βWYL*, а если *γk*= 1, то *βwylcd*= *βCD*, то

*k k k k*

Достаточное приличное условие уже доказано в [[16](#_bookmark48), [29](#_bookmark61)] и [[12](#_bookmark45)] соответственно. Теперь выведем достаточное условие спуска в случае 0*< γk<* 1. Из ([2.1](#_bookmark5))

*|βwylcd*|≤| *βWYL|*+*| βCD|*

*k*

2

+

gk+1

*gk*+1 *k k*

gk  *g *2



+1 *gk*

*k*

*gk*

2

≤ *g*

+ k+1

*|*− *g(T) d( |*

*k k)*

= + gk+1 *.* 2



2

gk+1

*gk*

2

2



*|*− *gTd|*

*k k*

Тогда из приведенного выше неравенства получаем

*Вайлд Т T*



2

gk+1

*gk*

2

2

*|*

gk+1 2 *T*

*βk gk*+1*dk|*≤ *|gk*+1*dk|*+*|*−*gTd | gk*+1*dk| ,* (3.2)



*| k k*

используя ([2.8](#_bookmark9)), получаем

*wylcd T T* 2



2

gk+1

*gk*

2

2

*|βk gk*+1*dk|*≤ *σ|g(k)dk|*+*σ * *g(k) (*+1) *,* (3.3)

Умножая ([2.2](#_bookmark6)) на *g(k) (*+1), находим

*gT dk*+1=−*gk*+12+*β(wylcd)gT dk.* (3.4)

k+1 *k* k+1

Затем *gT* dk+1 *wylcd gT dk*



gk+1

2

*(k) (*+1) =− 1+ *β*



gk+1 2

*k*

*(k) (*+1) *.* (3.5)

Теперь докажем свойство убывания направления *dk*по индукции. Имеем *gTd*1=− *g*12, предполагаем, что *di*, *i*= 1*,* 2*, . . . . , k* - все направления спуска (*gTdi<* 0).

1

*i* Из ([3.3](#_bookmark11)) следует, что

*wylcd T* 2 *gTdk* 2



*gk*

2

*|β g d* |≤− *2σ* *g*

*k* + *σ* *gk*(+1 *.* (3.6)

Тогда

*k k*+1 *k*

*k*+1

*2σ* *gk*+1

≤*βwylcdgT*

*k* k+1

*T)*

**(2) g( *k*



*gk*

2

*d(*

*k) d(k)*

2

*- σ gk*+1

≤− *2σ* *gk*+1

2 *gTdk* 2

+ *σ* *gk*(+1 *,* (3.7)



*gk*

2

*k*

Из ([3.5](#_bookmark12)) и ([3.7](#_bookmark13)) выводим

*gT* dk+1 *g(T) d(*

*k)*

*(k) (*+1) ≤− 1− *2σ k* + *σ.* (3.8)

 gk+12  *gk*2

Повторяя этот процесс

*gT* dk+1

k+1  gk+12

*g*

*(T) d(*

*k)*

≤− 1− *2σ k* + *σ*



*gk*

2

*gT dk 1*−

≤− 1− *2σ*( 1−− *2σ k*−1 + *σ*)+ *σ*



*gk 1*−

2

*gT dk 2*−

≤− 1+ *2σ*+ (*2σ*)2( 1−− *2σ k*−2 + *σ*)− *2σ*2 + *σ*



*gk 2*− 2

*gT dk 3*−

≤− 1+ *2σ*− (*2σ*)2− (*2σ*)3( 1−− *2σ k*−3 + *σ*)+ *σ*(*2σ*)2− *2σ*2+ *σ*



*gk 3*− 2

*gT*

*dk 4*−



*k*−4

*gk 4*−

2

≤− 1+ *2σ*− (*2σ*)2 + (*2σ*)3 + (*2σ*)4( 1−− *2σ*

−*2σ*2+ *σ*

.

+ *σ*)− *σ*(*2σ*)3+ *σ*(*2σ*)2

≤− 1+ *2σ*− (*2σ*)2+ (*2σ*)3− (*2σ*)4+ *. . .*

*Tk*− *(k*− *dk*− *(k*− 1)

−(*2σ*)*k*−1( 1−− *2σ*

*g*1) + *σ*)− *σ*(*2σ*)3+ *σ*(*2σ*)2− *2σ*2+ *σ*

*gk*− *(k*− 1) 2

*g(T) d(*1)

≤− 1+ *2σ*− (*2σ*)2+ (*2σ*)3− (*2σ*)4+-*- -*+ (*2σ*)*k*−1+ (*2σ*)*k* 1 − *σ*(*2σ*)*k*−1



*g*1

2

−*σ*(*2σ*)3+ *σ*(*2σ*)2− *2σ*2+ *σ,* (3.9)

использ1уя *gTd* =−*g *2, неравенство ([3.9](#_bookmark14)) становится

1 1

*gT* dk+1

*(k) (*+1) ≤− 1+ [(*2σ*+ (*2σ*)(3)+-*- -*+ (*2σ*)*k*−1)− ((*2σ*)2+ (*2σ*)4+-*- -*+ (*2σ*)*k*)]

gk+1 2

+[(*σ*+ *σ*(*2σ*)2+-*- -*+ *σ*(*2σ*)*k*−2)− (*2σ*2+ *σ*(*2σ*)3+-*- -*+ *σ*(*2σ*)*k*−1)]

≤− 1+ [(*2σ*+ (*2σ*)3+-*- -*+ (*2σ*)*k*−1)+ ((*2σ*)2+ (*2σ*)4+-*- -*+ (*2σ*)*k*)]

+[(*σ*+ *σ*(*2σ*)2+-*- -*+ *σ*(*2σ*)*k*−2)+ (*2σ*2+ *σ*(*2σ*)3+-*- -*+ *σ*(*2σ*)*k*−1)]

*k k 1*−

Σ Σ

=− 1+

Σ

(*2σ*)*j*+ *σ*

j=1

*k*

(*2σ*)*j*

j=0

*k 1*−

Σ

у нас есть

=− 2+

∞

(*2σ*)*j*+ *σ*

j=0

(*2σ*)*j,* (3.10)

j=0

*k 1*− ∞

Σ*k* Σ Σ Σ

(*2σ*)*j<* (*2σ*)*j*= 1

1− (*2σ*)

и (*2σ*)*j<* (*2σ*)*j*= 1 *.*

1− (*2σ*)

j=0 j=0 j=0 j=0

Таким образом, неравенство ([3.10](#_bookmark15)) становится

*gT* dk+1 1+ *σ*

*(k) (*+1) ≤− (− )*,* (3.11)

 gk+1 2 1− (*2σ*)

принимая 0*< σ <*1, тогда

5

*T*

*g*k+1

dk+1

≤− (−

1+ *σ*

1− (*2σ*)

) gk+1

2*<* 0*.* (3.12)

Таким образом, по*k* индукции *gTdk<* 0 выполняется для всех *k*≥ 0.

Теперь мы докажем достаточное условие спуска *dk*. Если 0*< σ<*1, то достаточно, чтобы

возьмем *c*= 2−1+*(σ* 5

1−(2*σ*), где 0*< c<* 1. Тогда из неравенства ([3.12](#_bookmark16)) следует, что

*T*

*g*k+1 *dk*+1≤− *c* *gk*+1

2

*,* (3.13)

что указывает на выполнение достаточного условия спуска.

# Свойства конвергенции

Для глобальной сходимости предложенного алгоритма необходимы следующие предположения о целевой функции *f*.

**Предположение 1** ([[5](#_bookmark38)])**.** H1. *Множество уровней H*= *{x*∈ *Rn/f* (*x*)≤ *f* (*x*0)*}*

*ограничено, где x*0*- начальный вектор.*

H2. *В некоторой окрестности Q из H функция f непрерывно дифференцируема и ее градиент непрерывен по Липшицу, т.е. l*∃*>* 0 *такой, что*

 *g*(*x*)− *g*(*y*) ≤*l* *x*− *y* для всех*x, y*∈ *Q.* (4.1)

Из этих гипотез следует, что r∃¯*>*0, так что

 *g*(*x*) ≤*r , x*¯∀∈ *H.* (4.2)

**Лемма 1** ( [[8](#_bookmark41)])**.** *Пусть выполнены предположения H1 и H2 и рассматриваются методы, сформулированные в (*[*1.2*](#_bookmark2)*), (*[*1.4*](#_bookmark3)*), где {dk} - направление спуска, а αkвычисляется с помощью сильного линейного поиска Вульфа-Пауэла. Если*

Σ 1



*dk*

*зате*

*k 0*≥

=+∞ (4.3)

*м* lim *inf* *gk*=0*.* (4.4)

*k'*→∞

**Лемма 2** ( [[17](#_bookmark49)])**.** *Предположим, что вышеприведенные предположения H1 и H2*

*выполняются. Если dkнаправление спуска, а длина шага αkудовлетворяет условиям*

*зате м*

*T*

*g*k+1

*dk*≥ *σgTdkk, σ<* 1*,* (4.5)

1− *σ|d(T)gk|*

*αk*≥ *k .* (4.6)

*l*

**Доказательство.** Из ([4.5](#_bookmark21)), ([4.1](#_bookmark17)) следует, что:

*dk*2

- (1− *σ*)*gTdk*

*k*

≤*σgTdk*− *gTdk*

*k k*

≤*d(T) (g*

– *g* )

*k*

=*d(T) y(*

*k*+1 *k*

*k k)*

Тогда ([4.6](#_bookmark22)) выполняется.

≤ *dk* *y(k*

2

≤*lαk dk .*

Согласно лемме [2](#_bookmark20), условиям ([2.8](#_bookmark9)), ([3.1](#_bookmark10)) и ([4.2](#_bookmark18)) выводим, что *αk*

полученный в новом алгоритме wylcd, не 0, т.е. p∃*>* 0 такой, что

*αk*≥ *p,* для всех k ≥ 0*.* (4.7)

**Теорема 4.1.** *Предположим, что предположение* [*1*](#_bookmark0) *выполняется. Пусть последовательность {xk}k*≥0*генерируется предложенным "алгоритмом wylcd". Тогда*

lim *inf* *gk*=0*.* (4.8)

*k'*→∞

**Доказательство.** Предположим, что ([4.8](#_bookmark24)) ложно. Тогда существует *r>* 0, такое, что

 *gk> r.* (4.9)

Из ([4.9](#_bookmark25)) и ([3.1](#_bookmark10)) следует, что

2

*- gTkd* ≥ *c g* ≥ *cr*2*,* (4.10)

*k k*

1

−*g(T)* ≤

1

2)*.* (4.11)

*k (cr) (*

*d(*

*k)*

Зная, что *sk*= *xk*+1− *xk*, пусть *A*= max{ *x*− *y/x, y*∈ *H}* диаметр множества уровня *H*.т.е. *sk*≤*A*.

Из ([2.1](#_bookmark5)) и ([2.2](#_bookmark6)) следует, что

 *dk*+1≤*gk*+1+*| βwylcd| k* *dk* *.* (4.12)

Что касается ограниченности *βwylcd*, то*k* здесь возможны три случая. Если *γk*= 0, то

*βwylcd*= *βWYL*и если *γk*= 1, то *βwylcd*= *βCD*, эти два случая уже есть.

*k k k k*

доказано в [[16](#_bookmark48), [29](#_bookmark61)] и [[12](#_bookmark45)] соответственно.

Теперь, если 0*< γk<* 1

*|βwylcd|*=*|* (1− *γk*)*βWYL*+ *γk βCD|*

*k k k*

≤| *βWYL|*+*| βCD|*

*k k*

 gk+12+*gk*+1 gk



+1 *gk*

*k*

*gk*

2

≤ *g*



 *g *2

+ k+1

*|*− *g(T) d( |*

*k k)*

gk+1 2



2 gk+1

*gk*

2

2



= + *T .* (4.13)

*|*−*g d|k k*

Из ([4.9](#_bookmark25)), ([4.11](#_bookmark26)) и ([4.2](#_bookmark18)) следует, что

2 2

*|βwylcd|*≤2*r*¯ +*r*¯ = *E.* (4.14)

*k* r2 cr2

Согласно ([4.14](#_bookmark28)) и ([4.2](#_bookmark18)), соотношение ([4.12](#_bookmark27)) приобретает вид

 *dk*+1≤*r* ¯+*E* *dk* *.* (4.15)

Используя *d* =*sk*и исходя из ([4.7](#_bookmark23)), получаем

*k*

*αk*

*d* ≤*r*¯+*E*

*s(k*

≤ *r* ¯+*E A.*

Поэтому

k+1 *(α) (k) p*

Σ 1

=+∞ *.* (4.16)



*dk*

*k 0*≥

Теперь, применяя Лемму [1](#_bookmark19), мы приходим к выводу, что

lim *inf* *gk*=0

*k'*→∞

это противоречит ([4.9](#_bookmark25)), поэтому мы доказали ([4.8](#_bookmark24)).

# Численные эксперименты

В этом разделе мы опишем численные результаты нового предложенного алгоритма (алгоритма wylcd). Во-первых, мы численно доказываем, что предложенный алгоритм имеет квадратичную сходимость, во-вторых, мы анализируем эффективность предложенного алгоритма, сравнивая его с методами (WYL) и (CD). В последующих численных экспериментах все коды скомпилированы на ПК со характеристиками: Intel(R) Core(TM) i5-3210M CPU 2.50GHz 2.50 GHz, 4.00 Go RAM, используя профиль Dolan and Mor´e [[10](#_bookmark43)] в качестве инструмента оценки. Все алгоритмы используют сильные условия поиска линии Вульфа-Пауэла с параметрами *δ*= *0.*0001 и *σ*= *0.*1 и

завершаются, когда *gk*∞≤10−6.

## Квадратичная сходимость предложенного алгоритма

Как мы уже говорили в разделе [2,](#_bookmark4) метод Ньютона имеет квадратичную скорость сходимости, и это свойство очень полезно, поскольку означает быструю сходимость к оптимальному решению, поэтому мы построили новый гибридный алгоритм сопряженного градиента на основе направления Ньютона, с целью получить быструю сходимость метода Ньютона в численном смысле и достичь некоторых существенных особенностей алгоритма сопряженного градиента. Далее мы предлагаем новый численный тест, который демонстрирует квадратичную сходимость предложенного алгоритма, используя некоторые тестовые задачи, выбранные из [[7](#_bookmark40)] с размерностью n: 150*,* 250*,* 258*,* 360*,* 365*,* 1950*,* 2000*,* 2005, и для каждой итерации вычисляем коэффициенты ошибок *rk*с двумя последовательными итерациями

*rk*= (5.1)

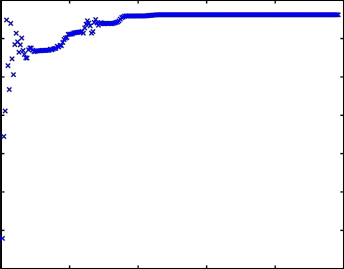


*xk*+1− *x*∗ *xk*− *x*∗ 2

где, *x*∗ - точное решение выбранной задачи. Мы построили график *зависимости log*(*rk*) от *k*

(см. рис. [1](#_bookmark29)).

**-0.5**



**0**

**50**

**100**

**150**

**200**

**250**

**-1**

**-1.5**

**-2**

**log( r )**

**k**

**-2.5**

**-3**

**-3.5**

**-0.5**

**Изображение выглядит как снимок экрана, черный, темнота

Контент, сгенерированный ИИ, может содержать ошибки.-1**

**-1.5**

**-2**

**log( r )**

**k**

**-2.5**

**-3**

**-3.5**

**-4**

**Число итераций k**

1. *n*= 150

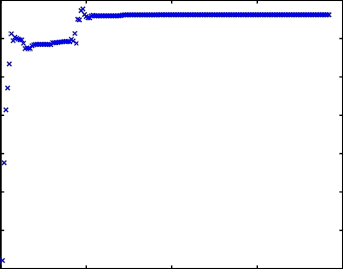
**-4**

**0 100 200 300 400**

**Число итераций k**

1. *n*= 250

**-0.5**



**0**

**50**

**100**

**150**

**200**

**-1**

**-1.5**

**-2**

**log( r )**

**k**

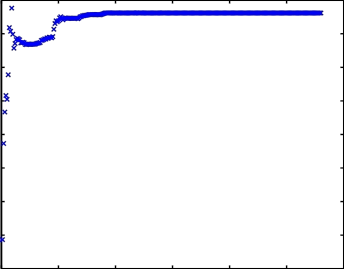
**-2.5**

**-3**

**-3.5**

**-0.5**

**-1**



**0**

**50**

**100**

**150**

**200**

**250**

**300**

**-1.5**

**-2**

**k**

**-2.5**

**log( r )**

**-3**

**-3.5**

**-4**

**-4**

**Число итераций k**

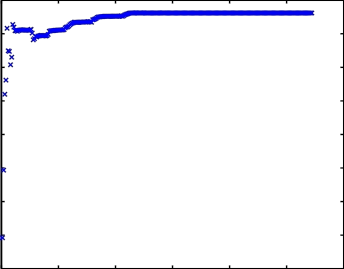
1. *n*= 258

**-4.5**

**Число итераций k**

1. *n*= 360

**-0.5 0**



**0**

**50**

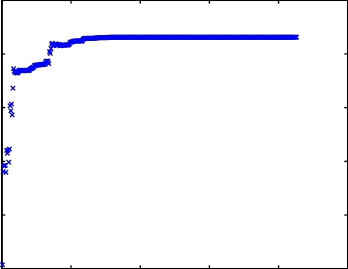
**100**

**150**

**200**

**250**

**300**



**0**

**100**

**200**

**300**

**400**

**500**

**-1**

**-1**

**-1.5**

**-2 -2**

**log( r )**

**k**

**log( r )**

**k**

**-2.5**

**-3 -3**

**-3.5**

**-4**

**-4**

**-4.5**

**Число итераций k**

1. *n*= 365

**-5**

**Число итераций k**

1. *n*= 1950

**Изображение выглядит как снимок экрана, пространство, темнота, черный

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как снимок экрана, пространство, Вселенная, темнота

Контент, сгенерированный ИИ, может содержать ошибки.0 0**

**-1 -1**

**-2 -2**

**log( r )**

**k**

**log( r )**

**k**

**-3 -3**

**-4 -4**

**-5**

**0 50 100 150 200 250 300**

**Число итераций k**

1. *n*= 2000

**-5**

**0 50 100 150 200 250 300**

**Число итераций k**

1. *n*= 2005

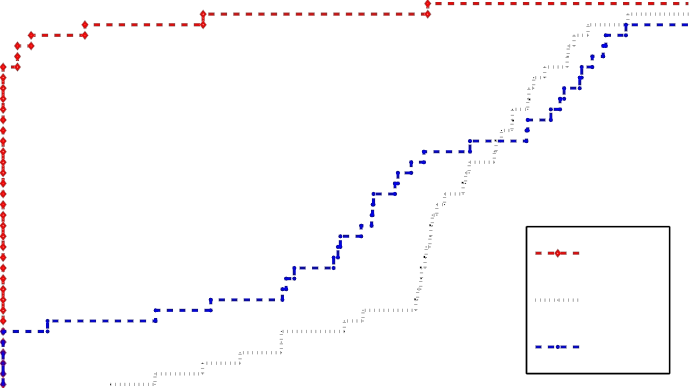
**Рисунок 1.** Квадратичная сходимость предложенного алгоритма

Как видно из рисунка [1](#_bookmark29), *log*(*rk*) *<* 0, т.е. *rk<* 1. Затем мы можем наблюдать, что *rk*имеет тенденцию сходиться к постоянному значению, то есть *rk*приближается к постоянному значению. Это наблюдение доказывает квадратичную сходимость нового алгоритма численном смысле.

## Числовые сравнения

Для оценки эффективности нашего "алгоритма wylcd" мы протестировали его в сравнении с алгоритмами (WYL) [[16](#_bookmark48), [29](#_bookmark61)] и (CD) [[12](#_bookmark45)], на основе которых он был построен, используя те же тестовые задачи, выбранные из [[7](#_bookmark40)], для каждой функции мы провели численные эксперименты для количества переменных: 2, 10, . . . , 10000. На рисунках [2](#_bookmark30), [3](#_bookmark31) и [4](#_bookmark32) приведены графики производительности этих алгоритмов по таким показателям: процессорное время, итераций, оценки градиента.

**1**



**wylcd**

**CD WYL**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**10**

**0.9**

**0.8**

**0.7**

**s**

**0.6**

**P(log (r )**≤ τ **:1**≤ **s**≤ **n )**

**2 p,s**

**0.5**

**0.4**

**0.3**

**0.2**

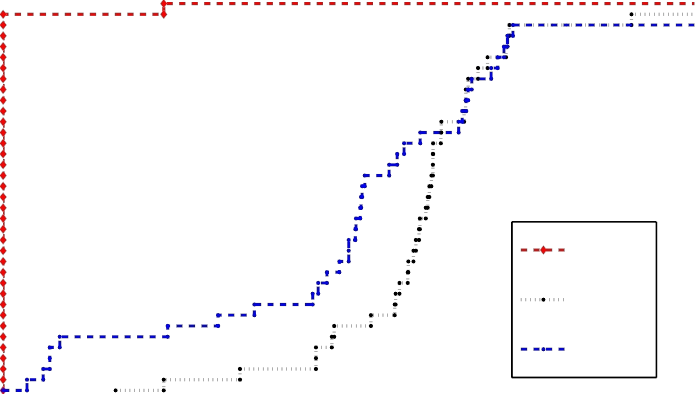
**0.1**

**0**

τ

**Рисунок 2.** Профиль производительности для процессорного времени

**1**



**wylcd CD**

**WYL**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9 10**

**0.9**

**0.8**

**0.7**

**s**

**0.6**

**P(log (r )**≤ τ **:1**≤ **s**≤ **n )**

**2 p,s**

**0.5**

**0.4**

**0.3**

**0.2**

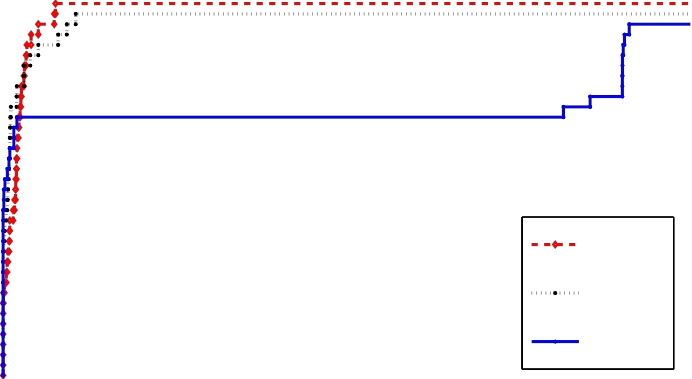
**0.1**

**0**

τ

**Рисунок 3.** Профиль производительности в зависимости от количества итераций.

**1**



**wylcd CD**

**WYL**

**0**

**5**

**10**

**15**

**20**

**25**

**0.9**

**0.8**

**0.7**

**s**

**0.6**

**P(log (r )**≤ τ **:1**≤ **s**≤ **n )**

**0.5**

**0.4**

**2 p,s**

**0.3**

**0.2**

**0.1**

**0**

τ

**Рисунок 4.** Профиль производительности для градиентных оценок.

Как видно из приведенных выше рисунков, новый алгоритм wylcd явно превосходит другие алгоритмы.

# Заключение

В этой научной статье мы представили новый алгоритм оптимизации, который объединяет некоторые существенные черты метода сопряженного градиента и ньютоновского метода, избегая при этом их недостатков. В частности, мы сосредоточились на сохранении простоты, низких требований к памяти, пригодности для решения задач с большой размерностью и глобальной сходимости метода сопряженного градиента. Одновременно мы стремились сохранить быструю квадратичную сходимость метода Ньютона в численном смысле, избегая при этом вычислительных затрат на прямое вычисление матрицы Гессиана и чувствительности сходимости к выбору начального вектора. Предлагаемый алгоритм представляет собой гибридный метод сопряженного градиента, в котором выпукло сочетаются методы (CD) и (WYL). По своей конструкции этот алгоритм близок к методу Ньютона без необходимости оценивать матрицу Гессиана за счет использования уравнения секущей, что делает его полезным для решения крупномасштабных задач оптимизации. Доказаны достаточное условие спуска и глобальная сходимость. Эффективность предложенного метода была проверена на наборе стандартных тестовых задач, что показало, что предложенный алгоритм приближается к квадратичной сходимости метода Ньютона и подтвердило его превосходство над алгоритмами (WYL) и (CD) в большинстве случаев.

# References

1. Z. M. Abdullah and I. K. Jamalaldeen, *A new hybrid of DY and CGSD con- jugate gradient methods for solving unconstrained optimization problems*, Tik.

J. of Pure Sci., 2021, 26(5), 86–91.

1. A. B. Abubakar, et al, *A Liu-Storey-type conjugate gradient method for uncon- strained minimization problem with application in motion control*, Journal of King Saud University - Science, 2022, 34(4), 101923.
2. A. B. Abubakar, P. Kumam, M. Malik, P. Chaipunya and A. H. Ibrahim, *A hybrid FR-DY conjugate gradient algorithm for unconstrained optimization with application in portfolio selection*, AIMS Mathematics, 2021, 6(6), 6506– 6527.
3. A. B. Abubakar, P. Kumam, M. Malik and A. H. Ibrahim, *A hybrid conjugate gradient based approach for solving unconstrained optimization and motion con- trol problems*, Mathematics and Computers in Simulation, 2022, 201, 640–657.
4. N. Andrei, *A hybrid conjugate gradient algorithm with modified secant condition for unconstrained optimization as a convex combination of Hestenes-Stiefel and Dai-Yuan algorithms*, Studies in Informatics and Control, 2008, 17(4), 373–392.
5. N. Andrei, *Another hybrid conjugate gradient algorithm for unconstrained op- timization*, Numer. Algor., 2008, 47(2), 143–156.
6. N. Andrei, *An unconstrained optimization test functions*, Collection. Adv. Model. Optim., 2008, 10(1), 147–161.
7. Y. H. Dai and Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM Journal on optimization, 1999, 10(1), 177– 182.
8. S. S. Djordjevi´c, *New conjugate gradient method as a convex combination of LS and FR methods*, Acta Mathematica Scientia, 2019, 39B(1), 214–228.
9. E. D. Dolan and J. J. Mor´e, *Benchmarking optimization software with perfor- mance profiles*, Math. Program., 2002, 91(2), 201–213.
10. N. J. Fanar and M. A. Ghada, *A new hybrid conjugate gradient algorithm as a convex combination of MMWU and RMIL nonlinear problems*, Journal of Interdisciplinary Mathematics, 2021, 24(3), 637–655.
11. R. Fletcher, *Practical Methods of Optimization. Unconstrained Optimization*, John Wiley and Sons, New York, 1987.
12. R. Fletcher and C. M. Reeves, *Function minimization by conjugate gradients*, Comput. J., 1964, 7(2), 149–154.
13. J. C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gra- dient methods for optimization*, SIAM J. Optimizat., 1992, 2(1), 21–42.
14. M. R. Hestenes and E. L. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand., 1952, 49(6), 409–436.
15. H. Huang, Z. Wei and S. Yao, *The proof of the sufficient descent condition of the Wei-Yao-Liu conjugate gradient method under the strong Wolfe-Powell line search*, Applied Mathematics and Computation, 2007, 189, 1241–1245.
16. J. K. Liu and S. J. Li, *New hybrid conjugate gradient method for unconstrained optimization*, Applied Mathematics and Computation, 2014, 245, 36–43.
17. Y. Liu and C. Storey, *Efficient generalized conjugate gradient algorithms, part 1: Theory*, J. Optim. Theory. Appl., 1991, 69(1), 129–137.
18. Y. Liu, Z. Zhu and B. Zhang, *Two sufficient descent three-term conjugate gradient methods for unconstrained optimization problems with applications in compressive sensing*, J. Appl. Math. Comput., 2022, 68, 1787–1816.
19. S. Narayanan and P. Kaelo, *A linear hybridization of Dai-Yuan and Hestenes- Stiefel conjugate gradient method for unconstrained optimization*, Numer. Math. Theor. Meth. Appl., 2021, 14, 527–539.
20. H. Oviedo, *Implicit steepest descent algorithm for optimization with orthogo- nality constraints*, Optimization Letters, 2022, 16(6), 1773–1797.
21. E. Polak and G. Ribi´ere, *Note sur la convergence de m´ethodes de directions conjugu´es*, Rev. Fr. Inf. Rech. Oper., 1969, 3(16), 35–43.
22. B. T. Polyak, *The conjugate gradient method in extremal problems*, USSR Com- put. Math. Math. Phys., 1969, 9(4), 94–112.
23. B. T. Polyak, *Newton’s method and its use in optimization*, European Journal of Operational Research, 2007, 181, 1086–1096.
24. A. Samson, *Nonlinear Programming: Theories and Algorithms of Some Un- constrained Optimization Methods (Steepest Descent and Newton’s Method)*, International Journal of Engineering and Management Research, 2020, 10(2), 1–12.
25. H. J. M. Shi, Y. Xie, R. Byrd and J. Nocedal, *A noise-tolerant quasi-Newton al- gorithm for unconstrained optimization*, SIAM Journal on Optimization, 2022, 32(1), 29–55.
26. Y. Wang, F. Alpak, G. Gao, C. Chen, J. Vink, T. Wells and F. Saaf, *An efficient bi-objective optimization workflow using the distributed quasi-Newton method and its application to well-location optimization*, SPE Journal, 2022, 21(1), 364–380.
27. H. A. Wasi and M. A. K. Shiker, *Nonlinear conjugate gradient method with modified armijo condition to solve unconstrained optimization*, J. Phys.: Conf. Ser., 2021, 1818, 1–7.
28. Z. Wei, S. Yao and L. Liu, *The convergence properties of some new conjugate gradient methods*, Applied Mathematics and Computation, 2006, 183, 1341–

1350.

1. X. P. Zhao, J. C. Yao and Y. Yao, *A nonmonotone gradient method for con- strained multiobjective optimization problems*, J. Nonlinear Var. Anal., 2022, 6(6), 693–706.