Численная оптимизация проверочной матрицы LDPC-кода для применения в протоколе квантового распределения ключей с использованием высокопараллельных вычислений*

В.И. Морозов, В.О. Башара, М.В. Емельяненко

Национальный исследовательский университет «Высшая школа экономики»
(101000, г. Москва, ул. Мясницкая, д. 20)

Исправление ошибок в секретном ключе является обязательным этапом протоколов квантового распределения ключей (КРК). Для его реализации, как правило, используются современные помехоустойчивые коды. Несовершенство аппаратуры, используемой в системах КРК, приводит к появлению битовых ошибок в канале. Более того, для подобных систем характерно несимметричное распределение таких ошибок. Учет такой асимметрии в модели канала не только позволяет повысить эффективность используемого кода, но и оставляет открытым вопрос об оптимальности существующих проверочных матриц для данной задачи. В данной работе проводится оптимизация проверочной матрицы LDPC-кода для заданного квантового канала. Полученные результаты показывают применимость численной оптимизации стандартных проверочных матриц для повышения эффективности КРК.

Ключевые слова: квантовое распределение ключей, помехоустойчивое кодирование, LDPC, оптимизация.

1. Введение

Развитие современных квантовых технологий открывает возможности для практической реализации систем, существование которых несколько десятилетий назад было только теорией. Одним из видов таких систем являются системы квантового распределения ключей (КРК). Протоколы КРК позволяют двум сторонам информационного обмена выработать общую последовательность бит. Благодаря тому, что секретность такой последовательности обеспечивается фундаментальными законами квантовой механики [1], биты могут быть использованы сторонами в качестве секретного ключа для дальнейшей защиты канала коммуникации.

Большинство протоколов квантового распределения ключей содержат схожий набор этапов, которые можно разделить на две основные группы: непосредственная передача квантовых состояний и алгоритмическая постобработка полученной «сырой» последовательности (подробно этапы протоколов КРК будут описаны в разделе 3). Помимо прочих этапов, обработка полученной последовательности бит, как правило, включает исправление битовых ошибок-инверсий, которые всегда имеют место в силу несовершенств практических реализаций протоколов КРК. Для эффективного согласования результирующей ключевой последовательности между сторонами используются методы помехоустойчивого кодирования.

Эффективность многих современных кодов, исправляющих ошибки (например, LDPC-кодов), во многом определяется структурой проверочной матрицы — прямоугольной матрицы над полем $GF(2)$, задающей набор проверок на четность. При этом для различных

*Публикация подготовлена в ходе проведения исследования 25-00-061 «Алгоритмическое обеспечение систем квантового распределения ключей» в рамках Программы «Научный фонд Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ)» с использованием суперкомпьютерного комплекса НИУ ВШЭ.

каналов связи наилучшую эффективность могут показывать разные проверочные матрицы. В силу того, что квантовая среда передачи, по которой производится отправка «сырого» ключа, представляет собой канал, принципиально отличный от наиболее распространенных (таких как, например, двоичный симметричный канал или двоичный канал с гауссовским шумом), выработка оптимальной проверочной матрицы для такого канала не является решенной задачей.

Таким образом, целью данного исследования является получение проверочной матрицы LDPC-кода, демонстрирующей высокую эффективность при использовании в протоколах КРК. Для решения подобных задач используются различные подходы. Одним из наиболее эффективных является численная оптимизация, которая и будет использована в данной работе. В ходе такой оптимизации проверочная матрица итерационно подвергается различным преобразованиям, не меняющим ее размерность. Те из преобразований, которые приводят к улучшению кода по заданному критерию качества, остаются в результирующей матрице. Удачное построение такой последовательности преобразований позволяет достигать значительного прироста корректирующей способности кода для заданных параметров канала.

Дальнейшее повествование организовано следующим образом. В разделе 2 приведен краткий обзор научных работ, посвященных построению и оптимизации проверочных матриц помехоустойчивых кодов, а также использованию таких кодов в протоколах КРК. Раздел 3 содержит описание работы протоколов КРК, применяемых в них методов исправления ошибок, а также используемого в данной работе подхода к оптимизации. Раздел 4 посвящен описанию программной реализации применяемых методов численной оптимизации. В разделе 5 представлены результаты вычислительных экспериментов по оптимизации проверочной матрицы. Обсуждение полученных результатов и выводы по проделанной работе приведены в разделе Заключение.

2. Обзор актуальных работ по тематике исследования

С момента появления квантовой криптографии в 1984 году [2] было предложено множество усовершенствований оригинального протокола BB84, а также немалое количество новых. Усовершенствования касались не только подходов к передаче квантовых состояний, но и алгоритмической постобработки ключа, в том числе и исправления ошибок. Первые протоколы, применяемые для восстановления ключевой информации в КРК [3,4], были основаны на проверках четности и, согласно утверждению авторов [4], были довольно неэффективны.

Результатом последующих исследований в этом направлении стал протокол CASCADE [5], также основанный на проверках четности и двоичном поиске. Этот протокол получил дальнейшее развитие в более поздних работах (например, [6, 7]), однако большую популярность в современных исследованиях получили методы помехоустойчивого кодирования, нашедшие свое применение как в КРК, так и во многих других областях. Такими методами являются полярные и LDPC-коды.

Так, работы [8, 9] посвящены применению однопроходных схем исправления ошибок в КРК. Такие схемы отличаются отсутствием необходимости во взаимодействии с передающей стороной во время исправления ошибок. Это позволяет увеличить скорость генерации ключа за счет уменьшения количества передаваемой информации. Авторы используют полярные и LDPC-коды и достигают лучших результатов, чем другие однопроходные подходы.

Большое количество современных работ [10–14] посвящено «слепому» исправлению ошибок в ключе. Такое название метод получил из-за того, что в нем не производится предварительная оценка ошибки в канале — QBER (quantum bit error rate). Вместо этого авторы предлагают использовать методы «выкалывания» и «укорачивания», чтобы в про-

цессе исправления ошибок итеративно подстроить скорость кода — отношение количества информационных бит к общему количеству бит кодового слова — таким образом, чтобы ее было достаточно для исправления имеющегося уровня ошибок. Большинство подходов, предложенных авторами подобных исследований, используют LDPC-коды и достигают результатов, близких к пределу Шеннона.

Следует отметить, что полярные коды, введенные позднее, чем LDPC, также нашли свое применение в КРК. Так, в работе [15] предлагается использование полярных кодов в КРК. А исследование [16], в свою очередь, посвящено усовершенствованию подхода к кодированию таким образом, что это позволяет снизить необходимую длину блока до 4 раз. Это дает важное преимущество, в виду того, что возможность использования коротких кодов критически важна в задачах квантовой криптографии, т.к. на большой длине линии ключевая информация генерируется медленно.

Если же возвращаться к применению LDPC-кодов в КРК, то, помимо распространенного «слепого» декодирования, некоторые работы [17, 18] также посвящены многомерным схемам КРК, где в качестве носителей информации используются не двоичные кубиты, а d-мерные «кудиты», несущие в себе более одного бита информации. Для исправления ошибок в таких схемах используются недвоичные LDPC-коды. Как утверждают авторы, такие подходы позволяют достигать большей эффективности исправления ошибок, чем классические двоичные.

Таким образом, исходя из содержания рассмотренных работ в области согласования информации в протоколах квантового распределения ключей, можно заключить, что LDPC-коды являются одним из наиболее популярных инструментов для решения данной задачи, т.к. предоставляют высокую эффективность, а также возможность адаптации к особенностям КРК. По этой причине именно такой тип кодов был взят в качестве объекта оптимизации в данной работе. Далее в обзорной главе пойдет речь об исследованиях, посвященных оптимизации LDPC-кодов.

В число базовых работ, направленных на построение и оптимизацию LDPC-кодов, которые нашли свое отражение во многих более новых исследованиях, входят статьи [19, 20].

В [19] описывается алгоритм Progressive Edge Growth (PEG), позволяющий выстраивать проверочную матрицу LDPC-кода псевдослучайным образом с соблюдением заданных ограничений. Основной метрикой качества кода, на которой и основан данный алгоритм, является охват графа Таннера (см., например, [21]) для данного кода — длина наименьшего цикла в таком графе. Данная метрика используется в качестве целевой функции во многих более современных исследованиях, т.к. может быть эффективно вычислена в ходе построения кода и хорошо коррелирует с его результирующей корректирующей способностью.

В свою очередь, авторы [20] описывают еще одну метрику качества кода — Extrinsic Message Degree (EMD), которая показывает степень связности цикла в графе с остальным графом через узлы переменных. Основываясь на этой метрике, авторы представляют подход ACE (Approximated Cycle EMD), позволяющий строить эффективные по EMD матрицы LDPC-кодов. Этот подход, а также его доработки и расширения (например, [22]), используется в качестве базового во многих дальнейших исследованиях.

Базируясь на рассмотренных выше работах, исследователи представляют различные подходы к оптимизации проверочных матриц. Так, ряд исследований [23–25] посвящен использованию генетических алгоритмов для решения данной задачи. В качестве целевой функции в данных работах тем или иным образом используются результаты симуляции исправления ошибок с использованием полученной на том или ином этапе оптимизации проверочной матрицы. В статье [23] авторы ищут точку пересечения графика зависимости выходного уровня ошибок (BER) от уровня шума в канале (SNR) с некоторым заданным значением. Удаленность этой точки от нуля берется в качестве значения целевой функции. В исследовании [24] в таких же целях используется мера способности полученного LDPC-кода исправлять все единичные ошибки в кодовом слове. А авторы [25], в свою очередь,

рассчитывают отношения отказов в декодировании к общему количеству попыток декодирования, произведенному во время симуляции (FER). При этом, в отличие от [23], расчет производится не на некотором диапазоне, а на единственном заранее заданном значении SNR.

Статьи других исследователей посвящены применению иных метаэвристических подходов к оптимизации LDPC-кодов. Так, в работе [26] предлагается использование метода роя частиц [27] для построения квазициклических LDPC-кодов. В качестве целевой функции авторы берут комплексный критерий, включающий проверку отсутствия циклов длины 4 в графе Таннера, а также значение выходного BER при заданном значении SNR. В более поздних работах авторы используют алгоритм имитации отжига [28] и метод дифференциальной эволюции [29] для построения более качественных кодов. Примечательно, что, т.к. последняя из перечисленных статей [29] направлена на оптимизацию энергопотребления декодера, в качестве целевой функции в ней используется не производительность кода, а временная и пространственная сложности декодирования.

Подходы к вычислению целевой функции, отличные от классических, встречаются и в других исследованиях. К примеру, авторы [30] используют комплексный многоступенчатый алгоритм в качестве критерия качества. Его вычисление включает как получение зависимости FER от SNR посредством симуляции, так и классические метрики, такие как ACE и охват графа. В более современной работе [31] используется сверточная нейронная сеть для оценки производительности получаемых в ходе оптимизации проверочных матриц. Авторы предлагают подход, включающий специальную предварительную обработку полученных кодов, а также дообучение сети в процессе оптимизации.

Наконец, ряд работ посвящен совершенствованию классических критериев и подходов к построению проверочных матриц. Так, в одной из работ последних лет [32] предлагается алгоритм GCE — Girth-Cycle-Embedding, позволяющий получать коды с любым заданным значением охвата. Суть алгоритма состоит в том, что узлы графа сначала собираются в циклы заданной длины, и только потом встраиваются в результирующий граф. Другое современное исследование [33] направлено на модификацию алгоритма CC-PEG (Cycle-concentrating progressive edge growth) [34], разработанного специально для эффективного расширения квазициклических кодов. Также, говоря о развитии классических подходов, следует упомянуть более раннюю работу [35], где авторы вводят метрику под названием «профиль охвата», на основе которой впоследствии осуществляется построение квазициклических LDPC-кодов с использованием жадного алгоритма.

Подводя итог, по результатам обзора исследований в данной сфере можно отметить, что, несмотря на популярность оптимизации LDPC-кодов среди исследователей, довольно малое внимание уделяется оптимизации протографов квазициклических кодов. Также зачастую авторы берут коды из современных телекоммуникационных стандартов исключительно для сравнения их по производительности со своими, построенными по заданным критериям, кодами. Что касается целевых функций, в виду ресурсозатратности вычислений, лишь в малой части рассмотренных работ в качестве метрики качества получаемой матрицы используются результаты симуляции декодирования на некотором диапазоне SNR (или входных BER). Таким образом, в данной работе были выбраны следующие подходы к решению задачи оптимизации проверочной матрицы LDPC-кода для использования в задачах квантового распределения ключей.

1. В качестве стартовой точки для оптимизации взята проверочная матрица из стандарта 5G [36].

2. В качестве целевой функции используется значение удаленности точки пересечения графика зависимости выходного FER от входного BER с пороговым значением, полученное в результате симуляции.

3. Теоретическая часть

3.1. Протоколы квантового распределения ключей

После того, как Беннет и Брассард в 1984 году представили первый протокол квантового распределения ключей [2], квантовая криптография ушла далеко вперед. Были разработаны как множество модификаций изначального протокола, так и другие схемы КРК, отличающиеся, например, другой передающей средой, другими носителями информации или использованием других свойств квантовых состояний для кодирования кючевой информации. Однако, несмотря на множество различий в физической реализации, с точки зрения алгоритмической постобработки информации, полученной из квантового канала, большинство протоколов имеют сходства. Во-первых, в большей части таких схем помимо квантового канала используется также классический открытый канал связи (например, Ethernet). Во-вторых, большая часть протоколов КРК включает следующие основные этапы.

1. *Передача квантовой информации.* Алиса (передающая сторона информационного взаимодействия) подготавливает квантовые носители информации и кодирует в каждом из них по одному или более информационных бит, сгенерированных случайно, выбирая базис кодирования также случайным образом. Недоступность базиса на первом этапе не позволяет потенциальному злоумышленнику провести качественное измерение информации, закодированной в квантовых носителях. Подготовленные носителя информации отправляются Бобу (принимающей стороне информационного взаимодействия). Последовательность информационных бит, полученная Бобом, называется сырым ключом. Все дальнейшие взаимодействия Алисы и Боба в рамках данной сессии протокола происходят только по открытому каналу.

2. *Просеивание ключа.* В процессе передачи по квантовому каналу большая часть носителей информации утрачивается в силу влияния внешних факторов. Боб определяет, какие именно из переданных Алисой квантовых носителей информации были приняты, а также (возможно, посредством дополнительного взаимодействия с Алисой) принимает решение о том, какие из них были измерены верно (например, в верном базисе). Сформированный список номеров верно принятых информационных единиц принимающая сторона отправляет передающей, после чего обе стороны отбрасывают все непринятые или неверно принятые приемником биты ключа. Полученная последовательность бит называется просеянным ключом.

3. *Оценка уровня ошибки в канале.* Помимо ошибок-стираний, когда передаваемый бит не доходит до получателя, квантовым каналам (в силу неидеальности приемо-передающей аппаратуры) также присуще наличие ошибок-инверсий, когда информационный бит принимается получателем, но при этом меняет свое значение на противоположное. На данном этапе Алиса и Боб производят оценку уровня таких ошибок в канале (QBER). Существует ряд различных подходов к оценке QBER, однако все они, так или иначе, требуют раскрытия части информации о секретном ключе. Раскрытые информационные биты отбрасываются на последнем этапе.

4. *Исправление ошибок.* В силу того, что полученные в результате работы протокола биты предназначаются для использования в качестве секретного ключа шифрования, даже при малом уровне QBER принимающей стороне необходимо исправить все ошибки-инверсии. Для этого, как правило, используются методы помехоустойчивого кодирования, позволяющие произвести исправление ошибок с раскрытием как можно меньшего (но не нулевого) количества информации о ключе. Более подробно этот этап будет рассмотрен далее в данной главе.

5. *Усиление секретности.* Чтобы результирующую последовательность можно было использовать в качестве секретного ключа шифрования, стороны проводят процедуру усиления секретности. Она состоит в хешировании полученной информации и отбрасывании того количества информации, которое было раскрыто в ходе предыдущих этапов. Результатом этого этапа является секретный ключ.

3.2. LDPC-кодирование

Как уже было отмечено ранее в данной главе, для исправления ошибок в секретных ключах в КРК используются методы помехоустойчивого кодирования [37]. Многие из таких методов для кодирования информации используют уравнения проверки на четность — линейные уравнения, включающие некоторые биты кодируемой информации, а также результат их сложения по модулю 2. Коэффициенты таких уравнений представляются в двоичном виде и собираются в проверочную матрицу H.

LDPC-коды [38] — семейство кодов, в котором используются проверочные матрицы с малым количеством единичных бит. В классическом кодировании, когда информация не является секретной, из проверочной матрицы H получают порождающую матрицу G. Умножение матрицы G на вектор информационных бит дает кодовое слово $c = mG$ — вектор бит, содержащий информационные биты, а также добавочные биты, содержащие дополнительную информацию, компенсирующую информационные потери при передаче слова по каналу связи. Матрица G подбирается таким образом, чтобы выполнялось $s = cH^T = 0$, где 0 обозначает вектор нулевых бит. Вектор s называют синдромом. В случаях же, когда кодовое слово искажено, такое перемножение с большой вероятностью дает ненулевой синдром, из чего стороны обмена могут сделать заключение о наличии искажений.

Чтобы восстановить искаженное кодовое слово $\hat{c} \neq c$, используются алгоритмы декодирования — итерационные алгоритмы, принимающие на вход оценки вероятностей значений каждого бита кодового слова (как правило, это логарифм отношения вероятности того, что бит равен 0, к вероятности того, что бит равен 1, — LLR — log likelihood ratio, $llr_i = log(p(c_i = 0)/p(c_i = 1))$), а также синдром, вычисленный для этого слова. По окончании работы, каждый из таких алгоритмов либо сообщает об успехе декодирования и возвращает битовый вектор $\widetilde{c}$, такой, что $\widetilde{c}H^T = 0$, либо сообщает об отказе в декодировании. Отношение количества отказов декодера к общему числу попыток декодирования различных искаженных кодовых слов называется FER (frame error rate) и является метрикой качества кода.

Таким образом, принцип работы классического LDPC-кодирования можно схематично представить так, как показано на рис. 1. Для восстановления исходного информационного вектора в классическом подходе используются такие декодеры как Sum-Product [39], Min-Sum [40, 41], а также различные их модификации.

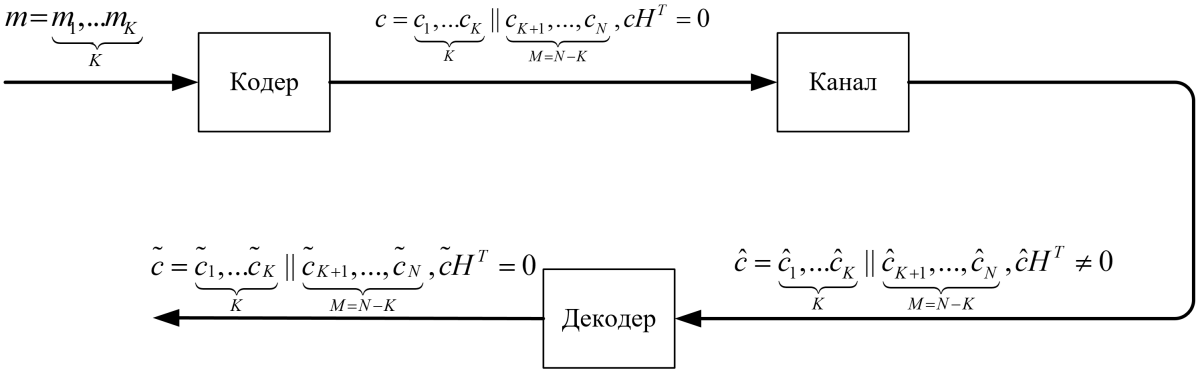

Рис. 1. Классическое помехоустойчивое кодирование

Однако классическая схема не подходит для квантового распределения ключей, т.к. требует передачи информационного вектора вместе с добавочными битами, что, в случае наличия злоумышленника, приведет к полному раскрытию секретного ключа. По этой причине в КРК используется альтернативный подход к кодированию [42]. В таком подходе Алиса передает Бобу вектор бит ключа m (по квантовому каналу), а также «синдром» $s = mH^T$. Для декодирования в таком случае используются модифицированные [43] версии классических декодеров, инвертирующие знак LLR на одном из этапов вычисления. Схема такого подхода к исправлению ошибок представлена на рис. 2.

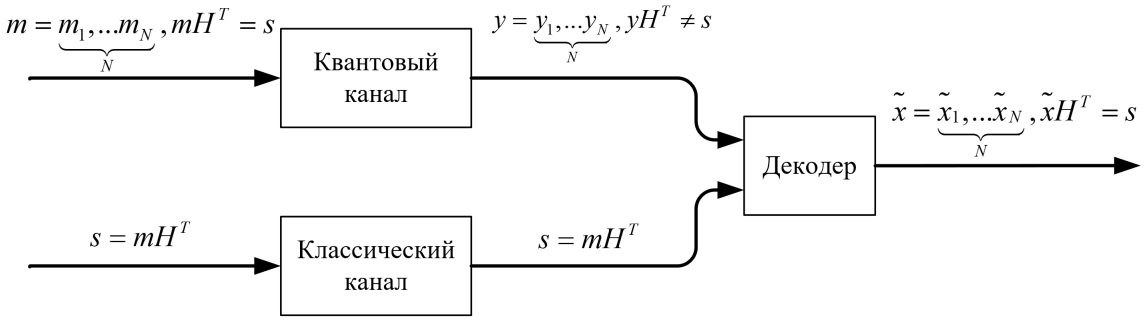

Рис. 2. Помехоустойчивое кодирование в КРК

3.3. Оптимизация проверочной матрицы — жадный алгоритм

Т.к. качество работы различных декодеров LDPC-кодов изменяется в зависимости от выбранной проверочной матрицы [21], можно утверждать, что эффективность того или иного LDPC-кода определяется его проверочной матрицей. Таким образом, в целях повышения эффективности исправления ошибок в секретном ключе можно рассматривать проверочную матрицу как объект оптимизации.

В современных телекоммуникационных стандартах (например, [36]) используются квазициклические коды. Проверочные матрицы таких кодов получаются из некоторой базовой матрицы BG, рекомендуемые варианты которых представлены в стандарте. Переход от базовой матрицы BG к проверочной матрице H производится заменой каждого единичного элемента в BG на квадратную единичную матрицу размера $Z_c \times Z_c$ с последующим циклическим сдвигом всех ее строк вправо на заданное количество позиций. Коэффициент расширения Z_c и длины циклических сдвигов для каждого из бит BG приводятся в стандарте.

Таким образом, несмотря на то, что большинство распространенных подходов к оптимизации LDPC-кодов направлены на построение или преобразование непосредственно самой матрицы H, в данной работе предлагается оптимизация базовой матрицы. Такой способ позволяет сократить количество итераций оптимизации за счет малого размера базовой матрицы, а также провести дополнительную оптимизацию сдвигов при необходимости. Значение целевой функции в данном случае вычисляется по следующему алгоритму.

На вход алгоритм принимает базовую матрицу BG, коэффициент расширения Z_c и количество итераций S, от которого зависит точность применяемого метода Монте-Карло.

1. Установить значение $i = 0$.
2. Расширить базовую матрицу BG на коэффициент Z_c, не производя циклические сдвиги единичных подматриц. Результат взять как матрицу H'.
3. Произвести циклический сдвиг тех из единичных подматриц размера $Z_c \times Z_c$, для которых размер сдвига не описан в стандарте, на случайное количество позиций. Результат взять как матрицу H_i.

4. Получить оценку качества матрицы $quality(H_i)$ (алгоритм будет описан далее).
5. Увеличить i на 1.
6. Если $i > S$, вернуть среднее значение метрики оценки качества по всем матрицам H_i, $i = \overline{0, S-1}$, иначе перейти к шагу 2.

Функция *quality* вычисляет оценку качества проверочной матрицы посредством симуляции по методу Монте-Карло. Функция принимает на вход следующие параметры.

- Двоичная матрица H размером $M \times N$.
- Величины симулируемой ошибки в канале e_{start}, e_{stop} и e_{step}.
- Величины, описывающие асимметричное распределение ошибок в канале E_i, каждая из которых соответствует одному из наблюдаемых в канале уровней ошибок.
- Величины, отвечающие за количество статистических итераций, определяющие точность метода Монте-Карло N_{total}, $N_{failures}$, I.

Оценка качества проверочной матрицы представляет собой вектор вещественных чисел, представляющих значения FER, соответствующие значениям $QBER$ в диапазоне от e_{start} до e_{stop} с шагом e_{step}. Вычисление этих значений производится по следующему алгоритму.

1. Установить $fers = ()$, где () обозначает пустой вектор вещественных чисел.
2. Установить значение текущей ошибки $e = e_{start}$.
3. Установить счетчик $i = 0$.
4. Установить $fer_{sum} = 0$.
5. Установить счетчики $t = 0, f = 0$.
6. Сгенерировать битовый вектор $m, |m| = N$, где $||$ — оператор получения длины вектора.
7. Рассчитать синдром $s = mH^T$.
8. Сгенерировать искаженный битовый вектор $y, |y| = N$, путем инверсии бит вектора m с вероятностью $e + E_k$, где $k \in \{1, 2, 3, 4\}$ выбирается равновероятно для каждого бита сообщения m.
9. Передать векторы s и y на вход декодеру LDPC-кода.
10. Если декодирование прошло успешно, перейти к шагу 12, иначе перейти к шагу 11.
11. Увеличить счетчик отказов f на 1.
12. Увеличить счетчик попыток декодирования t на 1.
13. Если $t < N_{total}$ и $f < N_{failures}$, перейти к шагу 6, иначе перейти к шагу 14.
14. Увеличить значение fer_{sum} на f/t.
15. Увеличить i на 1. Если $i < I$, перейти к шагу 5, иначе перейти к шагу 16.
16. Добавить значение fer_{sum}/I в вектор $fers$.
17. Увеличить e на e_{step}. Если $e \leq e_{stop}$, перейти к шагу 3, иначе вернуть вектор $fers$.

Под средним качеством матриц $H_0, H_1, ..., H_S$, качество каждой из которых определяется векторами $q_0, q_1, ..., q_S$ понимается вектор

$$q_{av} = (\sum_{i=0}^{S} q_i^0/S, \sum_{i=0}^{S} q_i^1/S, ..., \sum_{i=0}^{S} q_i^l/S),$$

где q_i^j — j-ый элемент i-ого вектора качества, а l — длина векторов качества ($l = \frac{e_{stop}-e_{start}}{e_{step}}$).

Алгоритм оптимизации принимает на вход базовую матрицу BG, коэффициент расширения Z_c, а также количество итераций оптимизации I_{opt}. Сам алгоритм представляет следующую последовательность шагов.

1. Расширить базовую матрицу BG размера $M_{BG} \times N_{BG}$ на коэффициент Z_c, произведя циклические сдвиги единичных подматриц согласно стандарту 5G [36]. Результат обозначить как матрицу H_{init}.

2. Получить вектор качества $q_{curr} = quality(H_{init})$.

3. Установить значение счетчика $i = 0$.

4. Сгенерировать два случайных числа $r, c \in \mathbb{N} : 0 < r < M_{BG}, 0 < c < N_{shift}$, где N_{shift} — индекс самого правого столбца базовой матрицы, для битов которого, согласно стандарту, производится ненулевой циклический сдвиг после развертывания в единичную подматрицу.

5. Инвертировать бит базовой матрицы BG в строке с номером r и столбце с номером c.

6. Рассчитать значение целевой функции для BG. Полученный вектор качества обозначить как q_i.

7. Если $q_i > q_{curr}$ (правила сравнения векторов качества будут описаны далее), установить $q_{curr} = q_i$ и перейти к шагу 9, иначе перейти к шагу 8.

8. Отменить изменения матрицы BG, сделанные на текущей итерации.

9. Увеличить счетчик i на 1. Если $i > I_{opt}$, вернуть матрицу BG, иначе перейти к шагу 4.

Неотъемлемым этапом многих алгоритмов оптимизации является сравнение значений целевых функций. Т.к. в нашем случае значение целевой функции представляет собой вектор, необходимо дополнительно привести правила сравнения таких значений. Для этого прежде всего фиксируется некоторое малое пороговое значение $fer_{threshold}$ представляющее собой целевое значение для всех элементов вектора качества. Далее сравнение двух векторов качества q_1 и q_2 длины l производится по следующим правилам.

1. Если $\forall 0 \leq i < l : q_1^i < fer_{threshold}, q_2^i < fer_{threshold}$, (т.е. если оба вектора качества изобразить на координатной плоскости, взяв их элементы как значения по оси ординат, а значения по оси абсцисс установить в соответствующие значения QBER, то они будут лежать под прямой, параллельной оси абсцисс, проведенной на высоте $fer_{threshold}$), то качество вектора q_i рассчитывается как $\sum_{j=0}^{l-1} fer_{threshold} - q_i^j$.

2. Если же условие $\forall 0 \leq i < l : q^i < fer_{threshold}$ выполняется только для одного из векторов качества (т.е. только один вектор лежит под линией $fer_{threshold}$), то этот вектор считается более качественным.

3. Если же ни один из векторов не лежит под линией $fer_{threshold}$, то более качественным признается тот, который пересекает линию $fer_{threshold}$ правее по оси абсцисс.

Выполнение ситуаций 2 и 3 продемонстрировано на рис. 3. Графики 1 и 2 демонстрируют ситуацию 3, когда ни один из векторов качества не лежит полностью под линией $fer_{threshold}$. В данном случае качество вектора на графике 2 выше, т.к. его значения пересекают пороговую линию правее по оси абсцисс. График 3 демонстрирует ситуацию 2, когда один из векторов полностью расположен под линией $fer_{threshold}$. Согласно пункту 2 критериев сравнения, вектор качества, изображенный на графике 3, демонстрирует лучшее качество, чем векторы на графиках 1 и 2.

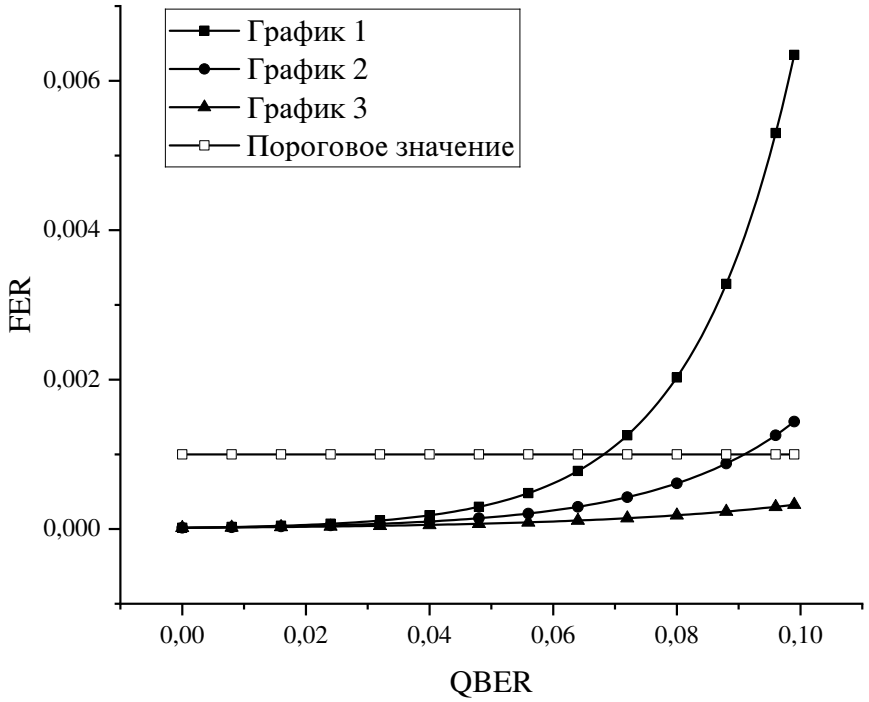

Рис. 3. Пример векторов качества

4. Программная реализация

Для исполнения алгоритмов, описанных в разделе 3, была разработана их программная реализация. С целью ускорения вычислений параллелизация производилась с использованием интерфейса MPI (Message Passing Interface), что позволило наиболее эффективно задействовать вычислительные узлы суперкомпьютерного комплекса НИУ ВШЭ [44]. При каждом запуске из всех параллельно работающих процессов выделялся один процесс-менеджер, остальные процессы назначались исполнителями. В ходе работы алгоритм оптимизации выполнялся процессом-менеджером. Однако расчет значений целевой функции делегировался процессам-исполнителям. Значения целевой функции для матриц с различными случайными сдвигами единичных подматриц независимы друг от друга. Благодаря этому, их вычисление производилось параллельно несколькими процессами-исполнителями, что привело к значительному приросту производительности оптимизации по сравнению с последовательным вариантом. Процессы-исполнители, в свою очередь, использовали многопоточность для вычисления качества проверочной матрицы на конкретных значениях QBER. Таким образом, ролевая структура MPI-процессов в программном исполнении оптимизатора была организована, как показано на рис. 4. N на рисунке означает количество одновременно запускаемых процессов-исполнителей. Содержимое сообщений «Задача» и «Результат» описано далее в данной главе.

Программная реализация оптимизатора исполнена на языке C++ с использованием следующего набора библиотек.

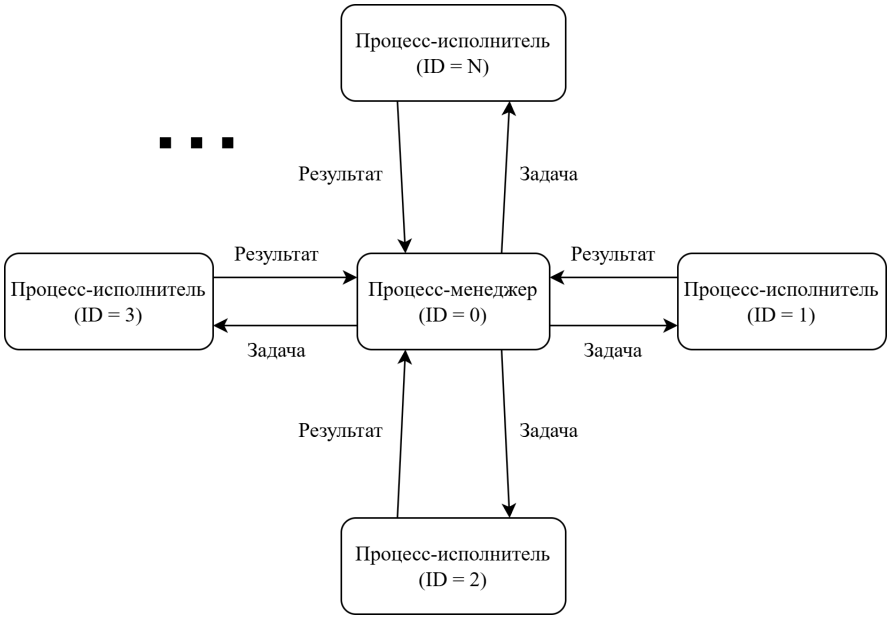

Рис. 4. Ролевая структура MPI-процессов

- OpenMPI с доступом через интерфейс Boost::MPI — для управления параллельными вычислениями и передачей сообщений между процессами по протоколу MPI.
- Eigen — для работы с матрицами, векторами и другими средствами линейной алгебры.
- Taskflow — для эффективной организации параллельных вычислений в пределах одного процесса.

Диаграмма классов представлена на рис. 5. Подробности, связанные с синхронизацией потоков, опущены для упрощения. Как видно из диаграммы, в программе выделены 4 основные сущности.

1. Класс Manager представляет менеджер задач, который получает задачи на вычисление целевой функции с теми или иными параметрами, распределяет их по процессам-исполнителям и собирает результаты.
2. Класс Worker представлет исполнителя, который получает параметры вычисления целевой функции, производит расчет и отправляет результат менеджеру.
3. Класс Task представляет входные параметры целевой функции (сообщение «Задача» на рис. 4).
4. Класс Result представляет результат вычисления целевой функции — вектор качества матрицы (сообщение «Результат» на рис. 4).

Сериализация объектов классов Task и Result для передачи между процессами производится средствами библиотеки Boost. При этом для экономии сетевых ресурсов проверочная матрица представляется в виде вектора позиций единичных вхождений, что, благодаря малому количеству единиц в ней, позволяет значительно сэкономить объем пересылаемого трафика. Более того, разреженность матрицы позволила отводить 1 байт информации для хранения каждого ее бита, а также осуществить совместное использование потоками элементов ее компактного представления с целью локализации данных на процессах-исполнителях.

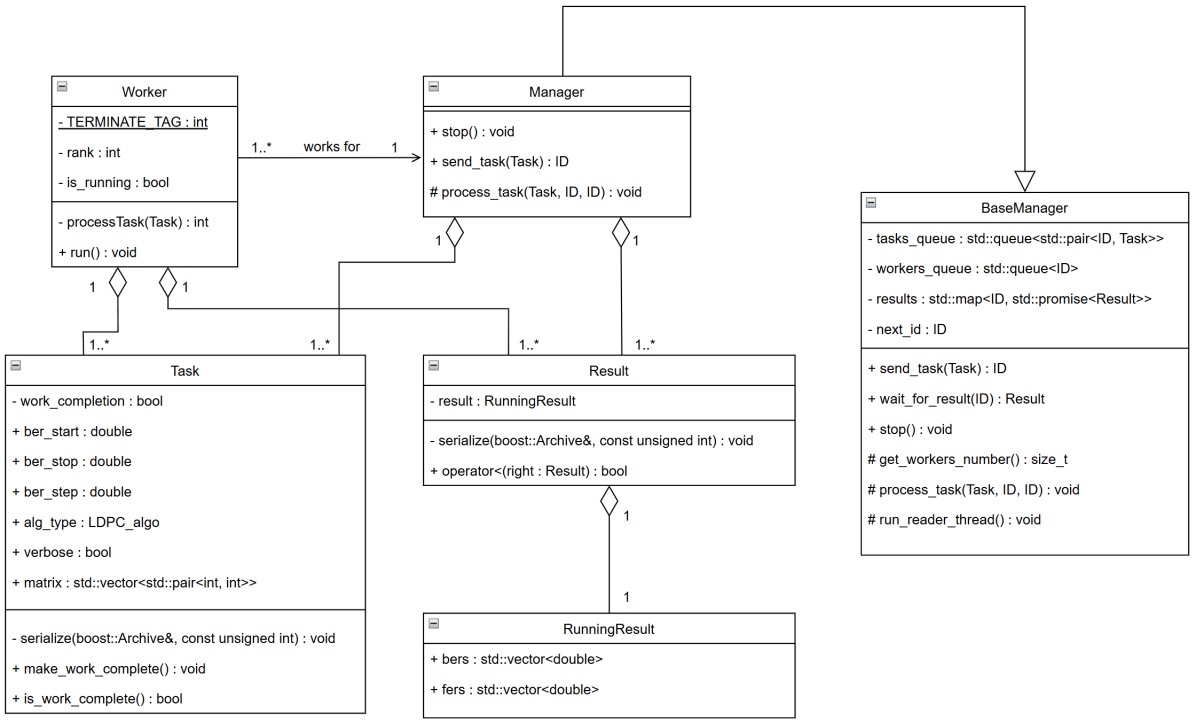

Рис. 5. Диаграмма классов

Оператор сравнения класса Result позволяет использовать обычную синтаксическую конструкцию для определения лучшего (по описанным выше критериям) из двух векторов качества проверочной матрицы.

Этап вычисления целевой функции для набора матриц, отличающихся только сдвигами единичных подматриц в их составе, происходит по следующему простому алгоритму.

1. Процессы-исполнители запускаются при старте программы и вызывают метод run класса Worker, который ожидает поступления задач и приступает к вычислению результатов.

2. Процесс-менеджер осуществляет сдвиги единичных подматриц и передает задачи вычисления целевой функции для них через метод send_task класса Manager.

3. Процесс-менеджер ожидает результатов вычислений через метод wait_for_result класса Manager.

4. Процесс-менеджер рассчитывает среднее значение метрики оценки качества.

Объектно-ориентированная реализация позволяет с незначительными доработками использовать представленное ПО для организации произвольных параллельных вычислений с использованием как MPI, так и других библиотек. Наследуя основные классы программы, можно контролировать выполнение вычислений на необходимом уровне от изменения целевой функции и ее параметров до смены механизма многозадачности.

5. Эксперимент

Оптимизация проверочной матрицы была проведена согласно алгоритму, описанному в подразделе 3.3. Для этого были установлены следующие значения для параметров алгоритмов.

1. В качестве базовой матрицы BG была выбрана матрица «Base graph 1» из стандарта 5G [36]. При этом она была усечена до $M_{BG} = 22$ строк и $N_{BG} = 44$ столбцов для получения скорости кода $R = 1 - M/N = 1/2$.

2. Коэффициент расширения Z_c был взят равным 4, что в итоге дает длину кодируемого блока $N = 44 \times 4 = 176$ бит. Такая длина была выбрана по двум причинам. Во-первых, выбранное значение позволило уменьшить сложность декодирования и, как следствие, выполнить больше итераций оптимизации. Во-вторых, что более важно, в системах квантового распределения ключей при большой длине линии скорость генерации просеянного ключа достаточно мала, из-за чего исправление ошибок на малом количестве бит является типичной задачей.

3. Количество сдвигов S было выбрано равным 10.

4. Величины e_{start}, e_{stop} и e_{step} были выбраны равными 0, 0.03 и 0.001 соответственно, что дает длину вектора качества $l = 30$.

5. Величины ошибок в канале E_i были выбраны следующими: $E_1 = 0.005$, $E_2 = 0.01$, $E_3 = 0.2$, $E_4 = 0.4$, что дает средний QBER 0.01875. Такое количество уровней ошибок и значение среднего QBER соответствуют, например, величинам, наблюдаемым при работе протокола KPK с фазово-временным кодированием [37].

6. Значения величин, отвечающих за точность расчетов методом Монте-Карло были выбраны следующими: $N_{total} = 10000$, $N_{failures} = 50$, $I = 30$.

7. Количество итераций оптимизации I_{opt} было выбрано равным 100.

8. Пороговое значение для сравнения векторов качества $fer_{threshold}$ было установлено равным 0.001.

В результате проведенной оптимизации значения целевой функции изменились, как показано на рисунке 6. На оси абсцисс в приведенном графике указано значение среднего QBER в декодируемом блоке, рассчитанное по формуле $(\sum_{i=1}^{4} E_i)/4 + e$, где e — текущее значение добавляемой ошибки на каждой итерации вычисления целевой функции. На оси ординат показан средний уровень FER, полученный при декодировании входных векторов с заданным уровнем ошибки.

Как видно на графике, в результате оптимизации было достигнуто заметное улучшение FER на всем диапазоне вычисления целевой функции, что свидетельствует о большей эффективности применения оптимизированной матрицы для исправления ошибок в рассмотренном диапазоне QBER.

Чтобы продемонстрировать ход улучшения качества проверочной матрицы в процессе оптимизации, была использована метрика эффективности исправления ошибок в KPK, рассчитываемая по следующей формуле.

$$f_{EC} = \frac{M}{Nh(e)},$$

где M — длина синдрома, равная количеству строк в проверочной матрице, N — длина кодируемого блока, равная количеству столбцов в проверочной матрице, e — QBER, исправляемый кодом с заданной вероятностью (в нашем случае это $fer_{threshold} = 0.001$), а h — функция Шенноновской двоичной энтропии.

$$h(x) = -xlog_2(x) - (1-x)log_2(1-x)$$

Данная метрика показывает отношение количества информации, фактически раскрываемого при исправлении ошибок, к теоретическому минимуму количества информации, которое

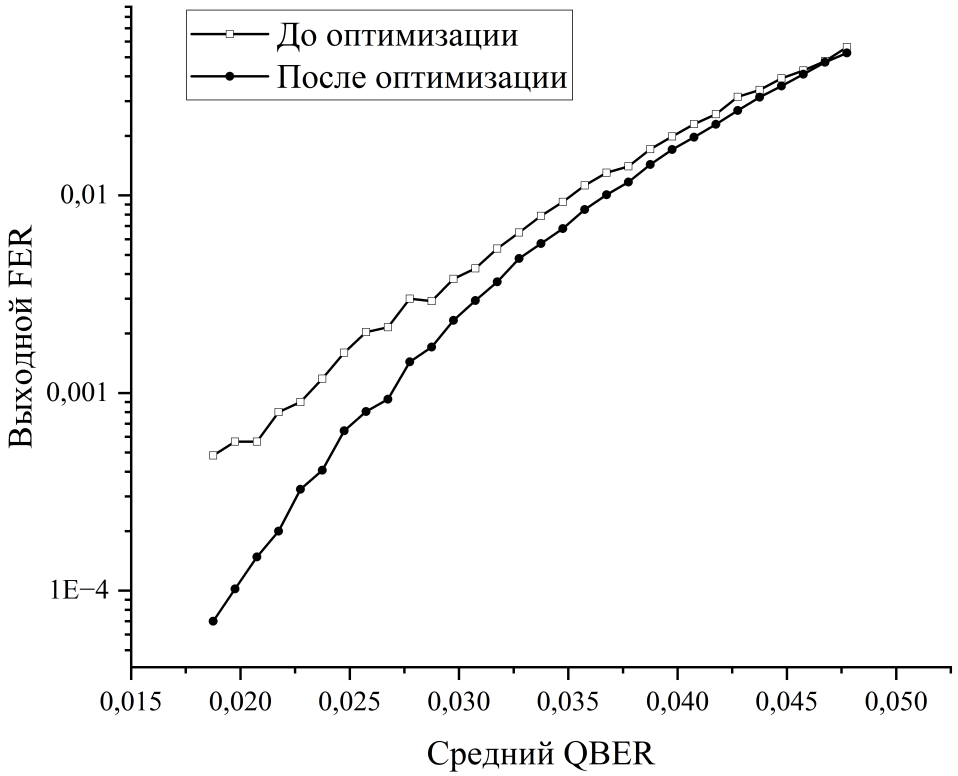

Рис. 6. Значения целевой функции до и после оптимизации

требуется раскрыть для исправления заданного уровня ошибок. Очевидно, $f_{EC} \geq 1$, где равенство достижимо лишь теоретически. Следовательно, чем меньше достигнутое значение f_{EC}, тем больше эффективность кода.

На рис. 7 представлена зависимость значения f_{EC} от номера итерации оптимизатора. При этом крайнее левое значение соответствует эффективности исправления ошибок на стартовой точке до оптимизации. Как видно, в ходе 100 итераций оптимизации f_{EC} было уменьшено примерно на 0.3.

Таким образом, по результатам применения представленного алгоритма оптимизации можно судить о его применимости для повышения корректирующей способности LDPC-кода в протоколах квантового распределения ключей. Также следует отметить, что наибольшие улучшения были достигнуты на первых 50 итерациях оптимизации, что свидетельствует о возможности остановки алгоритма или смены оптимизирующего преобразования после данного количества итераций.

6. Заключение

Таким образом, в ходе проведенного исследования был предложен алгоритм оптимизации проверочной матрицы LDPC-кода для использования в системах квантового распределения ключей. Согласно результатам вычислительных экспериментов, предложенный подход позволил улучшить эффективность декодирования на 0.3, а также достигнуть видимого улучшения на графике зависимости FER от QBER.

Более того, для реализации алгоритма было разработано ПО, модульная структура ко-

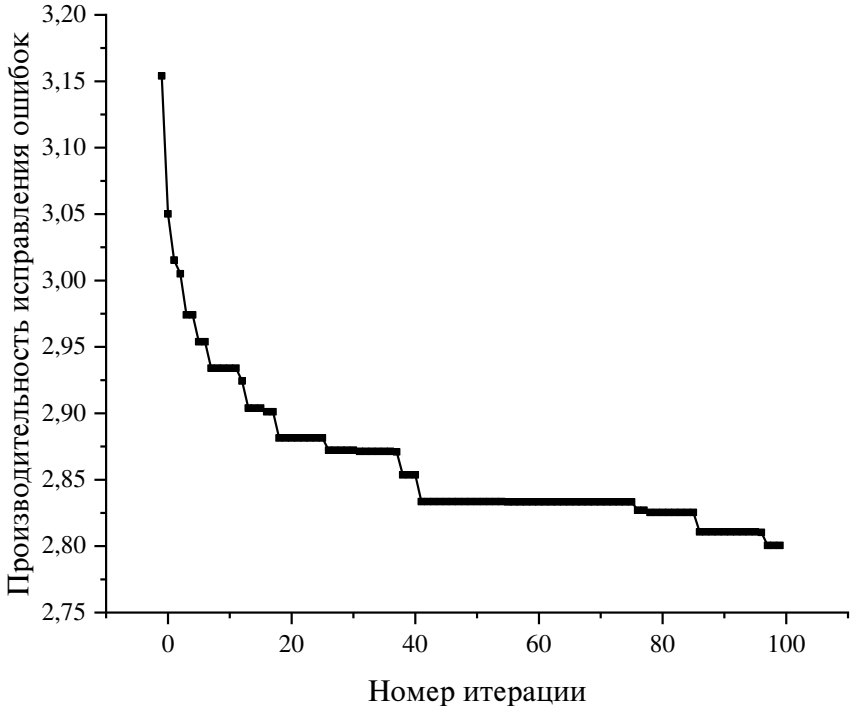

Рис. 7. Улучшение производительности декодирования в ходе оптимизации

торого предоставляет возможность реализации широкого круга параллельных вычислений.

Подводя итог по результатам проделанной работы, можно выделить следующие потенциальные направления дальнейших исследований.

1. Доработка примененного алгоритма оптимизации с использованием нескольких эпох, на каждой из которых инвертируется различное количество бит базовой матрицы.

2. Применение других подходов к оптимизации, к примеру, метаэвристических алгоритмов.

3. Применение более сложного подхода к вычислению целевой функции с использованием дополнительной оптимизации сдвигов единичных подматриц вместо их случайного выбора.

Литература

1. Wootters W.K., Zurek W.H. A single quantum cannot be cloned // Nature. 1982. Oct. Vol. 299, no. 5886. P. 802–803. DOI: 10.1038/299802a0.

2. Bennett C.H., Brassard G. Quantum cryptography: Public key distribution and coin tossing // Theoretical Computer Science. 2014. Vol. 560. P. 7–11. Theoretical Aspects of Quantum Cryptography – celebrating 30 years of BB84. DOI: 10.1016/j.tcs.2014.05.025.

3. Bennett C.H., Bessette F., Brassard G. et al. Experimental Quantum Cryptography // J. Cryptol. 1992. Vol. 5, no. 1. P. 3–28. DOI: 10.1007/BF00191318.

4. Bechmann-Pasquinucci H., Gisin N. Incoherent and coherent eavesdropping in the six-state protocol of quantum cryptography // Physical Review A. 1999. June. Vol. 59, no. 6. P. 4238–4248. DOI: 10.1103/physreva.59.4238.

5. Brassard G., Salvail L. Secret-Key Reconciliation by Public Discussion // Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings. Vol. 765 / ed. by T. Helleseth. Springer, 1993. P. 410–423. Lecture Notes in Computer Science. DOI: 10.1007/3-540-48285-735.

6. Toyran M. More efficient implementations of CASCADE information reconciliation protocol // 24th Signal Processing and Communication Application Conference, SIU 2016, Zonguldak, Turkey, May 16-19, 2016. IEEE, 2016. P. 161–164. DOI: 10.1109/SIU.2016.7495702.

7. Mao H.-K., Li Q., Hao P.-L. et al. RETRACTED ARTICLE: High performance reconciliation for practical quantum key distribution systems // Optical and Quantum Electronics. 2022. Feb. Vol. 54, no. 3. DOI: 10.1007/s11082-021-03489-4.

8. Yang L., Dong H., Li Z. One-way information reconciliation schemes of quantum key distribution // Cybersecur. 2019. Vol. 2, no. 1. P. 16. DOI: 10.1186/S42400-019-0033-Z.

9. Tomamichel M., Martínez-Mateo J., Pacher C., Elkouss D. Fundamental finite key limits for one-way information reconciliation in quantum key distribution // Quantum Inf. Process. 2017. Vol. 16, no. 11. P. 280. DOI: 10.1007/S11128-017-1709-5.

10. Liu Z., Wu Z., Huang A. Blind information reconciliation with variable step sizes for quantum key distribution // Scientific Reports. 2020. Jan. Vol. 10, no. 1. DOI: 10.1038/s41598-019-56637-y.

11. Limei G., Qi R., Di J., Duan H. QKD Iterative Information Reconciliation Based on LDPC Codes // International Journal of Theoretical Physics. 2020. Mar. Vol. 59, no. 6. P. 1717–1729. DOI: 10.1007/s10773-020-04438-9.

12. Kiktenko E.O., Malyshev A.O., Fedorov A.K. Blind Information Reconciliation With Polar Codes for Quantum Key Distribution // IEEE Commun. Lett. 2021. Vol. 25, no. 1. P. 79–83. DOI: 10.1109/LCOMM.2020.3021142.

13. Zhang J., Jin H., Xu H., Feng J. A Rate Compatible LDPC Scheme in the Quantum Key Distribution System // 2023 3rd International Conference on Information Communication and Software Engineering (ICICSE). IEEE, 04/2023. P. 51–55. DOI: 10.1109/icicse58435.2023.10211941.

14. Borisov N., Petrov I., Tayduganov A. Asymmetric Adaptive LDPC-Based Information Reconciliation for Industrial Quantum Key Distribution // Entropy. 2023. Vol. 25, no. 1. P. 31. DOI: 10.3390/E25010031.

15. Jouguet P., Kunz-Jacques S. High Performance Error Correction for Quantum Key Distribution using Polar Codes // Quantum Info. Comput. 2014. Vol. 14, no. 3–4. P. 329–338. DOI: 10.5555/2600508.2600516.

16. Zhou H., Tang B.-Y., Chen H. et al. Appending Information Reconciliation for Quantum Key Distribution // Phys. Rev. Applied. 2022. Vol. 18, no. 4. DOI: 10.1103/PhysRevApplied.18.044022.

17. Mueller R., Ribezzo D., Zahidy M. et al. Efficient Information Reconciliation for HighDimensional Quantum Key Distribution // Quantum Information Processing. 2024. Vol. 23. Article 195. DOI: 10.1007/s11128-024-04395-w.

18. Müller R., Bacco D., Oxenløwe L.K., Forchhammer S. Information Reconciliation for HighDimensional Quantum Key Distribution using Nonbinary LDPC codes // 12th International Symposium on Topics in Coding, ISTC 2023, Brest, France, September 4-8, 2023. IEEE, 2023. P. 1–5. DOI: 10.1109/ISTC57237.2023.10273570.

19. Hu X., Eleftheriou E., Arnold D. Regular and irregular progressive edge-growth tanner graphs // IEEE Trans. Inf. Theory. 2005. Vol. 51, no. 1. P. 386–398. DOI: 10.1109/TIT.2004.839541.

20. Tian T., Jones C.R., Villasenor J.D., Wesel R.D. Selective avoidance of cycles in irregular LDPC code construction // IEEE Trans. Commun. 2004. Vol. 52, no. 8. P. 1242–1247. DOI: 10.1109/TCOMM.2004.833048.

21. Johnson S. Iterative Error Correction: Turbo, Low-Density Parity-Check and RepeatAccumulate Codes. Cambridge University Press, 2010. Iterative error correction: turbo, low-density parity-check and repeat-accumulate codes. URL: https://books.google.ru/books?id=Ni71AAAAMAAJ.

22. Vukobratovic D., Djurendic A., Senk V. ACE Spectrum of LDPC Codes and Generalized ACE Design // Proceedings of IEEE International Conference on Communications, ICC 2007, Glasgow, Scotland, UK, 24-28 June 2007. IEEE, 2007. P. 665–670. DOI: 10.1109/ICC.2007.114.

23. Broulim J., Georgiev V., Moldaschl J., Palocko L. LDPC code optimization based on Tanner graph mutations // 2013 21st Telecommunications Forum Telfor (TELFOR). IEEE, 11/2013. P. 389–392. DOI: 10.1109/telfor.2013.6716251.

24. Broulim J., Davarzani S., Georgiev V., Zich J. Genetic optimization of a short block length LDPC code accelerated by distributed algorithms // 2016 24th Telecommunications Forum(TELFOR). IEEE, 11/2016. P. 1–4. DOI: 10.1109/telfor.2016.7818770.

25. Elkelesh A., Ebada M., Cammerer S. et al. Decoder-in-the-Loop: Genetic OptimizationBased LDPC Code Design // IEEE Access. 2019. Vol. 7. P. 141161–141170. DOI: 10.1109/ACCESS.2019.2942999.

26. Yang Y., Xiao Y. Construction of QC-LDPC codes based on PSO algorithm // 4th IET International Conference on Wireless, Mobile amp; Multimedia Networks (ICWMMN 2011). IEEE, 2011. P. 118–122. DOI: 10.1049/cp.2011.0971.

27. Kennedy J., Eberhart R. Particle swarm optimization // Proceedings of International Conference on Neural Networks (ICNN'95), Perth, WA, Australia, November 27 - December 1, 1995. IEEE, 1995. P. 1942–1948. DOI: 10.1109/ICNN.1995.488968.

28. Usatyuk V., Vorobyev I. Simulated Annealing Method for Construction of High-Girth QCLDPC Codes // 41st International Conference on Telecommunications and Signal Processing, TSP 2018, Athens, Greece, July 4-6, 2018. IEEE, 2018. P. 1–5. DOI: 10.1109/TSP.2018.8441303.

29. Yaoumi M., Dupraz E., Leduc-Primeau F., Guilloud F. Energy optimization of quantized min-sum decoders for protograph-based LDPC codes // Ann. des Télécommunications. 2020. Vol. 75, no. 11. P. 615–621. DOI: 10.1007/S12243-020-00804-0.

30. Bocharova I.E., Johannesson R., Kudryashov B.D. A unified approach to optimization of LDPC codes for various communication scenarios // 8th International Symposium on Turbo Codes and Iterative Information Processing, ISTC 2014, 18-22 August 2014, Bremen, Germany. IEEE, 2014. P. 243–248. DOI: 10.1109/ISTC.2014.6955122.

31. Xiao Z., Li L., Xu J., Sha J. Construction of protograph LDPC codes based on the convolution neural network // China Communications. 2023. May. Vol. 20, no. 5. P. 84–92. DOI: 10.23919/jcc.2023.00.012.

32. Gao C., Liu S., Jiang D., Chen L. Constructing LDPC Codes with Any Desired Girth // Sensors. 2021. Vol. 21, no. 6. P. 2012. DOI: 10.3390/S21062012.

33. Lee H., Kim J. Modified CC-PEG Algorithm for Protograph-Based QC-LDPC Codes Over Non-Uniform Channel // IEEE Access. 2024. Vol. 12. P. 173660–173669. DOI: 10.1109/ACCESS.2024.3502712.

34. Yun D.-Y., Kim J.-W., Kwak H.-Y., No J.-S. The Cycle-Concentrating PEG Algorithm for Protograph Generalized LDPC Codes // IEEE Access. 2023. Vol. 11. P. 57285–57294. DOI: 10.1109/ACCESS.2023.3284314.

35. Bocharova I.E., Hug F., Johannesson R., Kudryashov B.D. A greedy search for improved QC LDPC codes with good girth profile and degree distribution // 2012 IEEE International Symposium on Information Theory Proceedings. IEEE, 07/2012. P. 3083–3087. DOI: 10.1109/isit.2012.6284129.

36. ISO/IEC/IEEE International Standard - Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications // ISO/IEC/IEEE 8802-11:2012(E) (Revison of ISO/IEC/IEEE 8802-11-2005 and Amendments). 2012. P. 1–2798. DOI: 10.1109/IEEESTD.2012.6361248.

37. Sinilshchikov I.V., Molotkov S. Decoy States and Low-Density Parity-Check ErrorCorrecting Codes in Quantum Cryptography with Phase–Time Coding // Journal of Experimental and Theoretical Physics. 2019. Aug. Vol. 129, no. 2. P. 168–196. DOI: 10.1134/s1063776119070124.

38. Gallager R. Low-density parity-check codes // IRE Transactions on Information Theory. 1962. Vol. 8, no. 1. P. 21–28. DOI: 10.1109/TIT.1962.1057683.

39. MacKay D. Good error-correcting codes based on very sparse matrices // IEEE Transactions on Information Theory. 1999. Vol. 45, no. 2. P. 399–431. DOI: 10.1109/18.748992.

40. Fossorier M., Mihaljevic M., Imai H. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation // IEEE Transactions on Communications. 1999. Vol. 47, no. 5. P. 673–680. DOI: 10.1109/26.768759.

41. Emran A.A., Elsabrouty M. Simplified variable-scaled min sum LDPC decoder for irregular LDPC codes // 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC). 2014. P. 518–523. DOI: 10.1109/CCNC.2014.6940497.

42. Slepian D., Wolf J. Noiseless coding of correlated information sources // IEEE Transactions on Information Theory. 1973. Vol. 19, no. 4. P. 471–480. DOI: 10.1109/TIT.1973.1055037.

43. Liveris A., Xiong Z., Georghiades C. Compression of binary sources with side information at the decoder using LDPC codes // IEEE Communications Letters. 2002. Vol. 6, no. 10. P. 440–442. DOI: 10.1109/LCOMM.2002.804244.

44. Kostenetskiy P.S., Chulkevich R.A., Kozyrev V.I. HPC Resources of the Higher School of Economics // Journal of Physics: Conference Series. 2021. Jan. Vol. 1740, no. 1. P. 012050. DOI: 10.1088/1742-6596/1740/1/012050.

Параллельный алгоритм для численного исследования дозвуковых реагирующих течений в цилиндрической системе координат

М.С. Мустайкин, Е.Е. Пескова

Национальный исследовательский Мордовский государственный университет

Статья посвящена разработке параллельного вычислительного алгоритма на основе технологии MPI для численного исследования дозвуковых газовых потоков в трубчатом реакторе. Математическая модель исследуемого процесса представляет собой уравнения газовой динамики в цилиндрической системе координат. Вычислительный алгоритм построен с учетом дозвукового характера течения. Разработанная программа показала рост ускорения и эффективности с увеличением количества расчетных ячеек.

Ключевые слова: математическое моделирование, дозвуковые течения, уравнения газовой динамики, технология MPI.

1. Введение

В настоящее время математическое моделирование является одним из основных методов исследования химически активных сред, наблюдение которых зачастую недоступно в лабораторных экспериментах. В связи с активным развитием вычислительных систем открывается возможность использовать в расчетах более сложные математические модели и большие сетки для получения результатов, наиболее приближенных к реальным течениям газа в химических реакторах.

Задачей, стоящей перед авторами статьи, является создание вычислительного алгоритма и программы для исследования дозвуковых химически активных газопылевых сред в присутствии лазерного излучения, поглощаемого газом и каталитическими наночастицами, в цилиндрической системе координат. Выбор системы координат обоснован необходимостью исследования газопылевых течений в круглых трубах. Трехмерная геометрия необходима для отражения вихревых процессов, наблюдаемых в исследуемых течениях, реализации различных углов ввода газовой смеси в трубу, несимметричного ввода лазерного излучения, что зачастую является необходимым условием интенсификации химических процессов.

В работах [1, 2] представлена математическая модель и разработан вычислительный алгоритм для исследования двухфазных газопылевых потоков с химическими реакциями и лазерным излучением для случая осесимметричных течений. Исходная математическая модель требует специализированного подхода при построении вычислительного алгоритма. Для обхода вычислительной сложности, возникающей при расчете изменений характеристик дозвуковых реагирующих сред, в алгоритме использован принцип расщепления по физическим процессам, ориентированный на выделение блоков для расчета каждого физического процесса с подбором адекватного численного метода. Сочетание быстрых химических реакций и медленных газовых потоков обуславливает необходимость выделения решения уравнений химической кинетики в отдельный блок. Отдельные трудности вызывает учет малых изменений давления в потоке. В условиях малых чисел Маха эти изменения незначительны, однако, они являются причиной изменения вектора скорости. В математической модели давление представлено в виде суммы давления, постоянного в области, и динамической поправке к давлению, которое меняется в каждой точке пространства на каждом шаге по времени. Также требует необходимости разработка отдельных блоков для учета лазерного излучения и характеристик каталитических наночастиц.

Целью настоящей работы является построение параллельного вычислительного алго-

ритма для уравнений газовой динамики в цилиндрической системе координат, поскольку уже на этапе учета только конвективных потоков и многокомпонентности смеси встала необходимость использования технологий параллельных вычислений.

2. Математическая модель и вычислительный алгоритм

Рассмотрим систему уравнений газовой динамики в цилиндрической системе координат. Данная система получена из уравнений Навье—Стокса в приближении малых чисел Маха с помощью отбрасывания диссипативных членов [3, 4].

$$\frac{\partial U}{\partial t} + \frac{\partial F^{(1)}(U)}{\partial z} + \frac{1}{r}\frac{\partial (rF^{(2)}(U))}{\partial r} + \frac{1}{r}\frac{\partial F^{(3)}(U)}{\partial \phi} = Q. \tag{1}$$

Здесь

$$U = \begin{pmatrix} \rho Y_m \\ \rho u_z \\ \rho u_r \\ \rho u_\phi \\ \rho h \end{pmatrix}, \quad F^{(1)}(U) = \begin{pmatrix} \rho u_z Y_m \\ \rho u_z^2 + \pi \\ \rho u_z u_r \\ \rho u_z u_\phi \\ \rho h u_z \end{pmatrix}, \quad F^{(2)}(U) = \begin{pmatrix} \rho u_r Y_m \\ \rho u_z u_r \\ \rho u_r^2 + \pi \\ \rho u_r u_\phi \\ \rho h u_r \end{pmatrix},$$

$$F^{(3)}(U) = \begin{pmatrix} \rho u_\phi Y_m \\ \rho u_z u_\phi \\ \rho u_r u_\phi \\ \rho u_\phi^2 + \pi \\ \rho h u_\phi \end{pmatrix}, Q = \begin{pmatrix} R_m \\ 0 \\ (\rho u_\phi^2)/r \\ (-\rho u_r u_\phi)/r \\ 0 \end{pmatrix}.$$

Здесь ρ – плотность газа, Y_m – массовая доля компоненты газа, $m = \overline{1, M}$ – номер компоненты газа, M – количество компонент в газовой смеси, R_m – скорость образования или расхода m-ой компоненты смеси, u_z, u_r, u_ϕ – компоненты вектора скорости, $\pi = p - p_0$ – динамическое отклонение давления, p – давление, p_0 – давление, постоянное в области, R_m – скорость образования или расхода каждой компоненты смеси для случая наличия химических реакций.

Энтальпию газовой смеси можно определить из температуры смеси T и массовых долей ее компонент:

$$h(T, Y_m) = \sum_m Y_m h_m(T), \quad h_m(T) = \int_{T_{ref}}^{T} C_{pm}(T)\, dT + h_m^0. \tag{2}$$

Здесь $T_{ref} = 293.15\ K$, h_m^0 – энтальпия образования каждой компоненты газа, $C_{pm}(T) = a_0 + a_1 T + a_2 T^2 + a_3 T^3$ – удельная теплоемкость каждой компоненты смеси при постоянном давлении, заданная многочленом, аппроксимирующим известные термодинамические табличные данные в нужном интервале температур [5].

Для построения дискретной модели введём равномерную по каждому направлению сетку:

$$\Omega_h = \left\{ z_i = ih_z, r_j = jh_r, \phi_k = kh_\phi; i = \overline{1, N_z}, j = \overline{1, N_r}, k = \overline{1, N_\phi}; N_z h_z = L_z, N_r h_r = L_r, N_\phi h_\phi = L_\phi \right\}$$

При построении вычислительного алгоритма используем схему расщепления по физическим процессам:

1. На первом шаге решается система уравнений химической кинетики:

$$\begin{cases} \dfrac{\partial \rho Y_1}{\partial t} = R_1, \\ \dfrac{\partial \rho Y_2}{\partial t} = R_2, \\ \ldots \\ \dfrac{\partial \rho Y_M}{\partial t} = R_M. \end{cases} \tag{3}$$

Для решения данной системы используется подключаемый модуль RADAU5 [6]. В результате находятся промежуточные значения вектора $\widetilde{\rho Y_m}$, $m = \overline{1, M}$.

2. На втором шаге решается система уравнений (1) без учета вклада динамического отклонения давления и химических реакций:

$$\frac{U^{n+1}_{ijk} - U^{n}_{ijk}}{\Delta t} + \frac{\widetilde{F}^{(1)}_{i+1/2jk} - \widetilde{F}^{(1)}_{i-1/2jk}}{h_z} + \frac{r_{ij+1/2k} \cdot \widetilde{F}^{(2)}_{ij+1/2k} - r_{ij-1/2k} \cdot \widetilde{F}^{(2)}_{ij-1/2k}}{r_{ijk} \cdot h_r} + \frac{\widetilde{F}^{(3)}_{ijk+1/2} - \widetilde{F}^{(3)}_{ijk-1/2}}{r_{ijk} \cdot h_\phi} = Q_{ijk}. \tag{4}$$

Здесь $\widetilde{F}^{(1)}_{i+1/2jk}$, $\widetilde{F}^{(2)}_{ij+1/2k}$, $\widetilde{F}^{(3)}_{ijk+1/2}$ – дискретные потоки между i и $i+1$, j и $j+1$, k и $k+1$ ячейками соответственно, для расчета которых применяется поток Русанова [7]:

$$\widetilde{F}^{(1)}_{i+1/2jk} = 0.5 \left(\widetilde{F}^{(1)} \left(U^{+}_{i+1/2jk} \right) + \widetilde{F}^{(1)} \left(U^{-}_{i+1/2jk} \right) - \alpha_1 \left(U^{+}_{i+1/2jk} - U^{-}_{i+1/2jk} \right) \right),$$
$$\widetilde{F}^{(2)}_{ij+1/2k} = 0.5 \left(\widetilde{F}^{(2)} \left(U^{+}_{ij+1/2k} \right) + \widetilde{F}^{(2)} \left(U^{-}_{ij+1/2k} \right) - \alpha_2 \left(U^{+}_{ij+1/2k} - U^{-}_{ij+1/2k} \right) \right),$$
$$\widetilde{F}^{(3)}_{ijk+1/2} = 0.5 \left(\widetilde{F}^{(2)} \left(U^{+}_{ijk+1/2} \right) + \widetilde{F}^{(2)} \left(U^{-}_{ijk+1/2} \right) - \alpha_3 \left(U^{+}_{ijk+1/2} - U^{-}_{ijk+1/2} \right) \right).$$

$U^{+}_{i+1/2jk}$, $U^{-}_{i+1/2jk}$ – значения вектора переменных U слева и справа от грани между i и $i+1$ ячейками, для которой вычисляется поток $\widetilde{F}^{(1)}_{i+1/2jk}$, $U^{+}_{ij+1/2k}$, $U^{-}_{ij+1/2k}$ – значения вектора переменных U слева и справа от грани между j и $j+1$ ячейками, для которой вычисляется поток $\widetilde{F}^{(2)}_{ij+1/2k}$, $U^{+}_{ijk+1/2}$, $U^{-}_{ijk+1/2}$ – значения вектора переменных U слева и справа от грани между k и $k+1$ ячейками, для которой вычисляется поток $\widetilde{F}^{(2)}_{ijk+1/2}$.

Стабилизирующий член потока находится из выражений [8, 9]:

$$\alpha_1 = \max \left(\sqrt{\left(u^{+}_{z,i+1/2jk} \right)^2 + \left(u^{+}_{r,i+1/2jk} \right)^2 + \left(u^{+}_{\phi,i+1/2jk} \right)^2}, \sqrt{\left(u^{-}_{z,i+1/2jk} \right)^2 + \left(u^{-}_{r,i+1/2jk} \right)^2 + \left(u^{-}_{\phi.i+1/2jk} \right)^2} \right),$$

$$\alpha_2 = \max \left(\sqrt{\left(u^{+}_{z,ij+1/2k} \right)^2 + \left(u^{+}_{r,ij+1/2k} \right)^2 + \left(u^{+}_{\phi,ij+1/2k} \right)^2}, \sqrt{\left(u^{-}_{z,ij+1/2k} \right)^2 + \left(u^{-}_{r,ij+1/2k} \right)^2 + \left(u^{-}_{\phi,ij+1/2k} \right)^2} \right),$$

$$\alpha_3 = \max\left(\sqrt{\left(u^+_{z,ijk+1/2}\right)^2 + \left(u^+_{r,ijk+1/2}\right)^2 + \left(u^+_{\phi,ijk+1/2}\right)^2},\right.$$
$$\left.\sqrt{\left(u^-_{z,ijk+1/2}\right)^2 + \left(u^-_{r,ijk+1/2}\right)^2 + \left(u^-_{\phi,ijk+1/2}\right)^2}\right).$$

В результате находятся плотность ρ, концентрации компонент смеси Y_m, энтальпия h, предварительные компоненты вектора скорости u^*_r, u^*_z, u^*_ϕ, из решения уравнения (2) методом Ньютона находится температура T.

3. На третьем шаге рассчитывается динамическая поправка к давлению и проводится коррекция вектора скорости.

Для расчета динамической поправки к давлению используется выражение [3]:

$$\nabla \cdot \frac{1}{\rho^n} \nabla \pi^{n+1} = \frac{1}{\Delta t} \left(\nabla \cdot \vec{u}^* - \nabla \cdot \vec{u}^{n+1} \right). \tag{5}$$

Здесь $\vec{u}^{n+1} = \dfrac{1}{\rho_{\text{g}}} \displaystyle\sum_m \left(\frac{M_w}{M_{wm}} - \frac{h_m}{C_p T_{\text{g}}} \right) R_m$ – выражение для дивергенции вектора скорости [3], M_w, M_{wm} – молекулярная масса смеси и компоненты соответственно.

Для решения уравнения Пуассона (5) используется метод Якоби:

$$\frac{\pi_{i+1jk} - 2\pi_{ijk} + \pi_{i-1jk}}{h_z^2} + \frac{\pi_{ij+1k} - 2\pi_{ijk} + \pi_{ij-1k}}{h_r^2} + \frac{1}{r_{ijk}^2} \frac{\pi_{ijk+1} - 2\pi_{ijk} + \pi_{ijk-1}}{h_\phi^2} + \tag{6}$$

$$+ \frac{1}{r_{ijk}} \frac{\pi_{ij+1k} - \pi_{ij-1k}}{2h_r} = G_{ijk}, \quad G_{ijk} = \frac{\rho^n}{\Delta t} \left(\nabla \cdot \vec{u}^* - \nabla \cdot \vec{u}^{n+1} \right). \tag{7}$$

После нахождения динамической поправки к давлению корректируем вектор скорости:

$$\vec{u}_z^{n+1} = u_z^* - \frac{\Delta t}{\rho^n} \frac{\partial \pi^{n+1}}{\partial z}, \tag{8}$$

$$\vec{u}_r^{n+1} = u_r^* - \frac{\Delta t}{\rho^n} \frac{\partial \pi^{n+1}}{\partial r}, \tag{9}$$

$$\vec{u}_\phi^{n+1} = u_\phi^* - \frac{\Delta t}{\rho^n} \frac{1}{r} \frac{\partial \pi^{n+1}}{\partial \phi}. \tag{10}$$

$$\tag{11}$$

В результате находятся динамическое отклонение от давления π, компоненты вектора скорости u_r, u_z, u_ϕ.

3. Параллельная реализация алгоритма

Расчетная область представляет собой цилиндр, покрытый равномерной сеткой с числом ячеек G_Nz, G_Nr, G_Np по соответствующему направлению (рис. 1).

За основу параллельного алгоритма принята трехмерная геометрическая декомпозиция области. Область разбивается на sz_z частей по оси OZ, sz_r частей по оси OR и sz_p по OP с равным количеством ячеек в каждой из образовавшихся подобластей. Каждая подобласть будет обрабатываться параллельно, и решение будет производиться по одинаковой схеме на каждом вычислительном узле. Для сохранения однородности схемы вычисления, по всем границам подобласти вводятся по одному слою фиктивных ячеек. Значения газодинамических параметров в этих дополнительных ячейках определяются либо исходя из граничных условий, если область является приграничной, либо посредством межпроцессорных обменов с соседними областями. Общее количество параллельных процессов равно $sz\_z \cdot sz\_r \cdot sz\_p$.

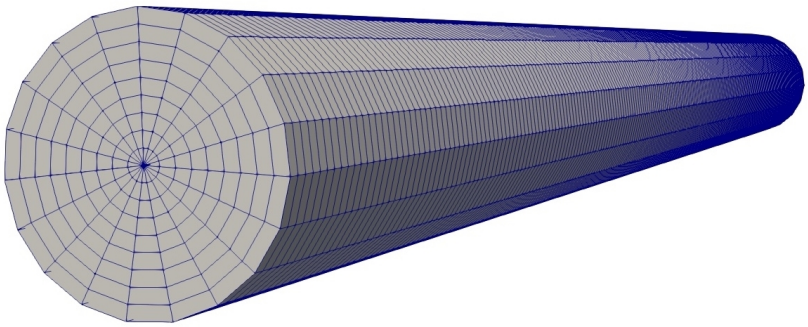

Рис. 1. Расчетная сетка

Параллельная программа разработана с использованием технологии для систем с распределенной памятью MPI, реализация механизмов межпроцессорного обмена осуществляется с помощью библиотеки MPICH. При запуске программы в первую очередь определяются соседи процесса по каждой границе и начальные координаты области на каждом параллельном процессе для возможности вычисления координаты любой ячейки. Начальных координат области достаточно, поскольку сетка равномерная по каждому направлению. Координаты ячейки необходимы для определения областей ввода смеси, лазерного излучения и вывода данных в файл. Далее, зная размерность сетки на каждом процессе $Nz \times Nr \times Np$, выделяется память под массивы газодинамических параметров и происходит их заполнение начальными данными. Далее вызывается процедура выполнения одного шага по времени, которая имеет следующую структуру:

1. Организации граничных условий в фиктивных ячейках. На исходной границе условия определяются обычным способом (условия втекания, вытекания, прилипания, отражения), в случае внутренней границы фиктивные ячейки заполняются посредством межпроцессорных обменов между соседними областями. Для пересылки данных в программе используются функции MPI_Send() и MPI_Recv().

2. Процедура расчета уравнений химической кинетики.

3. Процедура расчета уравнений газовой динамики.

4. Процедура решения уравнения Пуассона для динамической поправки к давлению. Поскольку используется метод Якоби, который требует небольшого количества итераций, внутри этой процедуры происходит межпроцессорный обмен до сходимости метода.

Вывод результатов производится каждым процессом в файлы с расширением *.vts, для визуализации используется программа ParaView.

4. Тестовый расчет

Параллельная версия предложенного алгоритма была реализована программно на языке C++. Для получения информации о производительности были выполнены расчеты для следующей задачи. Рассматривалась цилиндрическая труба диаметром 0.02 м и длиной 0.256 м. В начальный момент времени труба заполнена покоящимся метаном. Температура в области 600°C, давление 101325 Па. Слева в трубу поступает тоже метан, но с температурой 700°C и расходом 60 л/ч. На выходе из трубы справа задано условие вытекания, давление 101325 Па. Перепад давления на входе и выходе трубы задан 10^{-2} Па.

Поскольку нашей задачей является разработка основы параллельного алгоритма для дозвуковых двухфазных реагирующих течений и программа в дальнейшем будет расши-

ряться, температуры в начальных условиях приняты такие, что химические реакции разложения метана отсутствуют, смесь считается однокомпонентной. На рис. 2–3 представлены распределения плотности и температуры смеси, которые соответствуют физике процесса: с увеличением температуры в области посредством втекания горячего газа, плотность в этой области понижается.

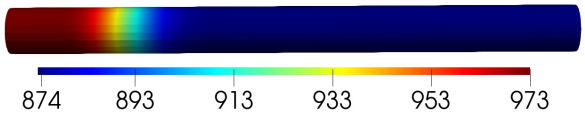

Рис. 2. Температура, K

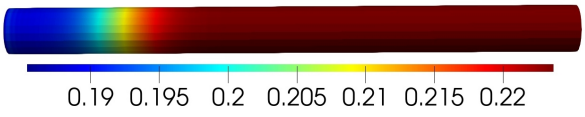

Рис. 3. Плотность, кг/м^3

Для оценки эффективности параллельного алгоритма измерялось время, затрачиваемое на выполнение определенного числа шагов по времени, с использованием различного количества параллельных процессов и расчетных ячеек. Расчеты проводились на рабочей станции ФГБОУ ВО «МГУ им. Н.П. Огарева» AMD Ryzen Threadripper 3990X 2900 МГц, 64 ядра.

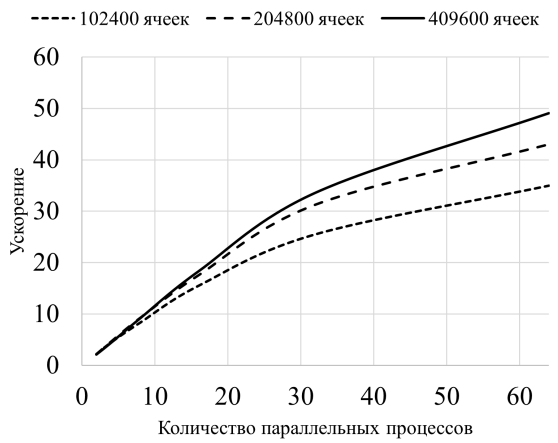

Рис. 4. Ускорение

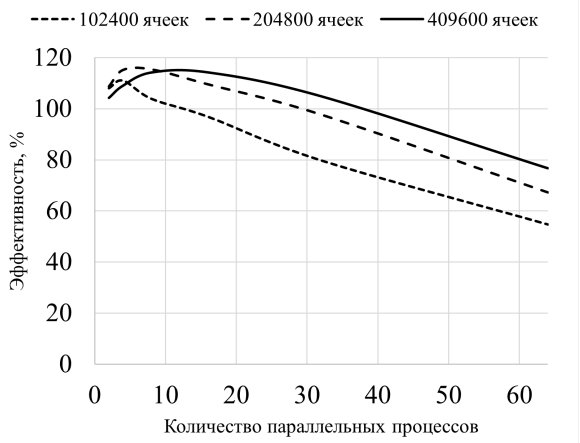

Рис. 5. Эффективность

При небольшом количестве параллельных процессов наблюдается суперлинейное ускорение и эффективность оказывается выше 100%. Однако, с увеличением количества процессов до 64 эффективность падает. Это связано с тем, что при увеличении количества декомпозиционных подобластей увеличивается и время, затраченное на межпроцессорные обмены. Из графиков видно, что с увеличением размерности расчетной сетки ускорение постепенно приближается к количеству вычислительных узлов и эффективность растет. Ожидается, что в случае включения громоздких схем химических реакций, которые будут независимо просчитываться на каждом процессе, эффективность на большом количестве процессов будет расти.

5. Заключение

Разработан параллельный вычислительный алгоритм с использованием технологии MPI для решения уравнений газовой динамики в цилиндрической системе координат с блочной структурой. Для разработанного параллельного алгоритма исследованы эффективность и ускорение для различного числа используемых процессоров. Сделаны выводы, что результирующий алгоритм позволит проводить серийные расчеты для различных углов ввода газовой смеси, а его блочная структура позволит расширить модель на учет различных эф-

фектов, возникающих в задачах моделирования химически активных двухфазных течений с лазерным излучением.

Литература

1. Snytnikov V.N., Peskova E.E., Stoyanovskaya O.P. Mathematical Model of a Two-Temperature Medium of Gas-Solid Nanoparticles with Laser Methane Pyrolysis // Mathematical Models and Computer Simulations. 2023. Vol. 15. P. 877–893. DOI: 10.1134/S2070048223050095.

2. Пескова Е.Е., Снытников В.Н., Жалнин Р.В. Вычислительный алгоритм для изучения внутренних ламинарных потоков многокомпонентного газа с разномасштабными химическими процессами // Компьютерные исследования и моделирование. 2023. Т. 15, № 5. С. 1169–1187. DOI: 10.20537/2076-7633-2023-15-5-1169-1187.

3. Day M.S., Bell J.B. Numerical simulation of laminar reacting flows with complex chemistry // Combustion Theory and Modelling. 2000. Vol. 4, no. 4. P. 535–556. DOI: 10.1088/1364-7830/4/4/309.

4. Борисов В.Е., Якуш С.Е. Применение адаптивных иерархических сеток для расчета течений реагирующих газов // Физико-химическая кинетика в газовой динамике. 2015. Т. 16, № 2. С. 1–13.

5. Stadnichenko O.A., Snytnikov V.N., Snytnikov Vl.N., Masyuk N.S. Mathematical modeling of ethane pyrolysis in a flow reactor with allowance for laser radiation effects // Chemical Engineering Research and Design. 2016. Vol. 109, P. 405–413. DOI: 10.1016/j.cherd.2016.02.008.

6. Hairer E., Wanner G. Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems. Springer-Verlag, Berlin, 1996.

7. Русанов В.В. Расчет взаимодействия нестационарных ударных волн с препятствиями // Журнал вычислительной математики и математической физики. 1961. Т. 1, № 2. С. 267–279.

8. Klein B., Muller B., Kummer F., Oberlack M. A high-order discontinuous Galerkin solver for low Mach number flows // International Journal for Numerical Methods in Fluids. 2015. Vol. 81, no. 8. P. 489–520. DOI: 10.1002/fld.4193.

9. Peskova E.E. Mathematical Modeling of Nonstationary Problems Related to Laser Thermochemistry of Methane in the Presence of Catalytic Nanoparticles // Doklady Mathematics. 2024. Vol. 109, no. 3. P. 256–261. DOI: 10.1134/S1064562424702107.