

О распараллеливании и визуализации при решении некорректных задач методами регуляризации и итерационной аппроксимации на вычислительном комплексе МВС-1000

В.В. Васин, Т.И. Серезникова, С.В. Шарф, П.А. Васев

Рассматриваются возможности применения многопроцессорной технологии и средств визуализации в процессе подбора параметров в методах регуляризации и итерационной аппроксимации при восстановлении разрывных решений для интегральных уравнений Фредгольма первого рода.

1. Введение

Решение многих прикладных задач сводится к решению операторных уравнений вида

$$Au = f, \quad (1)$$

где A - линейный оператор. Для устойчивой аппроксимации в общем случае разрывного решения рассматриваемых уравнений применяется двухэтапный регуляризующий алгоритм. На первом этапе методом регуляризации Тихонова операторное уравнение сводится к решению задачи минимизации

$$\min \{ \|Au - f\|^2 + \alpha \Omega(u - u^0) : u \in U \}, \quad (2)$$

где функционал $\Omega(u)$ играет роль стабилизатора. На втором этапе проводится дискретная аппроксимация регуляризованной задачи, а для решения последней применяется субградиентный метод, так как привлекаются негладкие стабилизирующие функционалы $\Omega(u)$.

Принципиальным вопросом при этом является выбор стабилизирующего функционала. Требования, предъявляемые к стабилизатору, носят в определенном смысле противоречивый характер. С одной стороны, этот функционал должен обладать достаточно сильным стабилизирующим эффектом, чтобы гарантировать сходимость, а с другой, не слишком заглаживать решение, чтобы не потерять тонкой структуры. Можно сказать, что проблема состоит в выборе подходящего стабилизатора и содержащихся в нем параметров и в выборе параметра регуляризации α .

В процессе численных экспериментов в первом случае восстанавливалось реальное двумерное изображение, искаженное регистрирующим прибором и аддитивной помехой. Во втором случае – восстанавливалось решение одномерного интегрального уравнения Фредгольма первого рода, при этом решение об изменении величины параметров, входящих в стабилизатор, принималось в процессе просмотра графиков градиентов и приближенных решений.

Для построения и вывода графиков использовалась специально разработанная подсистема онлайн-визуализации, которая позволяет пользователю наблюдать за текущим состоянием счета и управлять им. При визуализации осуществляется вывод графиков для текущей итерации, а также — вывод ряда параметров алгоритма. Управление заключается в приостановке счета, продолжении и выполнении ровно одной итерации. Дополнительно, пользователь может начать счет с новыми параметрами (без перезапуска самой программы на вычислителе).

2. Постановка задачи и регуляризация

Первая задача - восстановление изображения, искаженного аппаратной функцией K прибора наблюдения сводится к решению двумерного интегрального уравнения.

Как показано в работе [1], в этом случае в качестве стабилизирующего функционала можно использовать полную вариацию $J(u)$ в совокупности с нормой $L_2(D)$, $D \subset \mathbb{R}^2$,

$$\Omega(u) = \|u\|_{L_2(D)}^2 + \alpha J(u), D = \{0 \leq x \leq 1, 0 \leq y \leq 1\},$$

что позволяет обеспечить сильную сходимость регуляризованных решений в $L_2(D)$ и сходимость их вариаций.

Для дискретной аппроксимации регуляризованной задачи в единичном квадрате D строится равномерная сетка с шагом $h = 1/n$ по каждой переменной, т.е. вводится сеточный аналог \mathbb{R}_n^2 в пространстве \mathbb{R}^2 , $\mathbb{R}_n^2 = \{t \in \mathbb{R}^2 : t \in \{(ih, jh)\}, i, j = 0, \pm 1, \pm 2, \dots\}$, и сеточные функции $u_n : \mathbb{R}_n^2 \rightarrow \mathbb{R}$ с нормой

$$\|u_n\|_{l_2^n} = \sum_{t \in D_n} h^2 |u_n(t)|^2, \quad D_n = D \cap \mathbb{R}_n^2.$$

После некоторых преобразований функции $J_n(u_n)$ (суммирования по частям) экстремальная задача (2) преобразуется к виду:

$$\min \left\{ \sum_{k,l} \left[\sum_{i,j} h^2 K(y_k - t_i, y_l - s_j) u(t_i, s_j) - f_{k,l} \right]^2 h^2 + \alpha \sum_{i,j} h^2 \left[\left(\frac{u_{i,j} - u_{i,j-1}}{h} \right)^2 + \left(\frac{u_{i,j} - u_{i-1,j}}{h} \right)^2 \right]^{1/2} : u_n \in l_2^n \right\}. \quad (3)$$

Дискретный вариант субградиентного метода принимает вид:

$$u_n^{k+1} = u_n^k - \lambda_k \nabla \Phi_n^\alpha(u_n^k) / \|\nabla \Phi_n^\alpha(u_n^k)\|, \quad (4)$$

где $\nabla \Phi_n^\alpha(u_n^k)$ — произвольный субградиент целевой функции Φ_n^α из (6) в точке u_n ,

$$\lambda_k = (\Phi_n^\alpha(u_n^k) - \tilde{\Phi}_*^\alpha) / \|\nabla \Phi_n^\alpha(u_n^k)\|. \quad (5)$$

В формуле (4) компонента субградиента $\nabla \Phi_n^\alpha(u_n)$ имеет вид:

$$\begin{aligned} & 2 \sum_{k,l} h^2 \left[\sum_{i,j} K_{klij} u_{i,j} h^2 - f_{k,l} \right] K_{knlm} h^2 + \\ & + \alpha h^2 \left[\frac{(u_{nm} - u_{nm-1}) + (u_{nm} - u_{n-1m})}{\sqrt{(u_{nm} - u_{nm-1})^2 + (u_{nm} - u_{n-1m})^2}} - \right. \\ & - \frac{(u_{nm+1} - u_{nm})}{\sqrt{(u_{nm+1} - u_{nm})^2 + (u_{nm+1} - u_{n-1m+1})^2}} - \\ & \left. - \frac{(u_{n+1m} - u_{nm})}{\sqrt{(u_{n+1m} - u_{n+1m-1})^2 + (u_{n+1m} - u_{nm})^2}} \right]. \quad (6) \end{aligned}$$

Для продолжения экспериментов с выбором стабилизирующих функционалов была рассмотрена вторая задача (поставленная и теоретически обоснованная в работе [2]), в которой восстанавливается решение одномерного интегрального уравнения Фредгольма первого рода

$$Au \equiv \int_0^1 K(t, s) u(s) ds = f(t), \quad 0 \leq t \leq 1,$$

на основе тихоновской регуляризации с использованием стабилизатора в форме нормы Липшица. Вводится равномерная сетка по s, t с шагом $n = 1/N$, положим $u_n^0 = 0$. Тогда задачу можно редуцировать к следующей

$$\min \left\{ \max_{0 \leq i \leq N} \left| \sum_{j=0}^N \nu_j K(t_i, s_j) u_j - f_i \right| + \right.$$

$$+\alpha \left[\max_{0 \leq j \leq N} |u_j| + \max_{0 \leq i, j \leq N} \frac{|u_i - u_j|}{|t_i - t_j|^\mu} \right] : \{u_j\}_1^N \Big\},$$

где u_j — коэффициенты квадратурной формулы трапеций. Здесь субдифференциалы каждого из трех членов целевой функции вычисляются по известным правилам субдифференциального исчисления. Приведем, например, формулу для субдифференциала третьего слагаемого

$$\varphi(u) = \max_{0 \leq i, j \leq N} \frac{|u_i - u_j|}{|t_i - t_j|^\mu}.$$

Обозначим через $J(u)$ множество пар индексов i_0, j_0 , для которых в (9) достигается максимум.

Определим вектор

$$v_{i_0 j_0} = (0, \dots, 0, \frac{1}{|t_{i_0} - t_{j_0}|^\mu}, 0, \dots, 0, -\frac{1}{|t_{i_0} - t_{j_0}|^\mu}, 0, \dots, 0),$$

где ненулевые элементы занимают позиции с номерами i_0 и j_0 ; тогда

$$\partial\varphi(u) = \begin{cases} M = \text{co}\{v_{i_0 j_0}\}_{i_0, j_0 \in J(u)}, & \text{если } u_{i_0} - u_{j_0} > 0, \\ \text{co}\{M \cup (-M)\}, & \text{если } u_{i_0} - u_{j_0} = 0, \\ -M = \text{co}\{-v_{i_0 j_0}\}_{i_0, j_0 \in J(u)}, & \text{если } u_{i_0} - u_{j_0} < 0, \end{cases}$$

где $\text{co}\{M\}$ — выпуклая оболочка множества M , $u = (u_1, u_2, \dots, u_N)$.

3. Численные эксперименты

Решение задач проводилось на вычислительном комплексе МВС-1000 (для первой задачи более подробное изложение в [3]). Предварительный анализ формул в первой задаче показал, что на каждом шаге вычислений наибольшее время занимает получение субградиента. После нескольких экспериментов остановились на использовании шести процессоров. Больше число процессоров не использовалось, так как при увеличении количества процессов p число действий (оцениваемое как $O(n^4 p^{-1})$) в одном процессе уменьшается, а затраты на обмен "все-со-всеми", пусть слегка (по оценке $O(n^2 \log_2 p)$), но растут. Кроме того, увеличивается и вероятность попадания части процессов на менее связанные процессоры. Так как во время счета производилась настройка метода, резкое изменение длительности счета того же числа итераций, при выделении задаче менее связанных процессоров, приводил к потерям времени на выяснение того, что никаких ошибок в программу внесено не было.

В программе после нескольких итераций производится запись результатов в файл так, чтобы при следующем запуске можно было их прочитать и продолжить счет. Такая организация позволяет провести неограниченное число итераций и реализовать специфику основного вычислительного метода. Ввод-вывод возложен на нулевой процесс, который выполняет его наряду с вычислениями. Оказалось, что форматный вывод числовых данных не дает необходимой точности: младшие десятичные знаки после форматной записи и чтения искажаются. Пришлось реализовать бесформатное сохранение/восстановление, оставив выборочную форматную выдачу для слежения за ходом выполнения вычислений.

В целом же, перенос программы на МВС был произведен достаточно быстро, так как использовались простое разбиение по данным и, по сути, всего одна функция библиотеки MPI — `MPI_Allgather()`. Эта функция позволяет эффективно передать часть массива, полученную конкретным счетным процессом, всем другим и одновременно взять данные от них. Несмотря на то, что функция `MPI_Allgather()` — блокирующая, и с ее помощью не удастся совместить обмен с полезными вычислениями, основная причина существенного ускорения счета при описанном способе распараллеливания — в значительно большем, по сравнению с временем обмена, времени вычислений, как это видно из приведенных выше оценок. В процессе экспериментов при решении задачи о восстановлении изображения с применением

распараллеливания на шести процессорах был выбран подходящий параметр регуляризации $\alpha = 10^{-23}$, который обеспечил монотонное уменьшение как величины целевого функционала, так и величины абсолютной погрешности решения. Сходимость приближенных решений оказалась достаточно медленной: после выполнения примерно 12 тысяч итераций величина целевого функционала достигла значения 0.6×10^{-23} , норма абсолютной погрешности приближенного решения $\|\Delta u_n\|_{l_2^n} = 0.6 \times 10^{-5}$. Для получения изображения приближенных решений использовались графические возможности системы MatLab.

Для проведения расчетов во второй задаче о восстановлении решения одномерного интегрального уравнения прежде всего было необходимо выбрать параметры α и μ , входящие в расчетные формулы. Решение об изменении величины выбранных параметров в этом случае уже принималось в результате просмотра графиков текущего приближения к решению и градиента с выводом на эти же графики соответствующих числовых значений. Для этого использовалась специально разработанная подсистема визуализации, которая подключается и функционирует следующим образом.

В вычислительную программу вставляются несколько вызовов функций из библиотеки подсистемы, декларирующих адреса и символические имена данных, подлежащих визуализации. После запуска параллельной программы на рабочей машине пользователя запускается второй компонент подсистемы — визуализатор, который с помощью специального системного посредника связывается с выполняемой задачей и запрашивает требуемые данные. Запрос на данные в каждом процессе задачи обрабатывается в отдельном рабочем потоке, синхронизирующем доступ к памяти со счетным потоком с помощью дополнительных примитивов.

Сетевое взаимодействие между компонентами системы основано на протоколе SOAP, запросы осуществляются в формате XML, данные большого объема передаются с помощью двоичных приложений (DIME).

Компонент, реализующий визуализацию и интерфейс управления счетом данной задачи, основан на технологии интернет-программирования Flex 2, что позволяет применять визуализацию удаленно и в различных ОС.

На представленном ниже рис. 1 выведено приближенное решение для точного решения, приведенного на рис. 2, причем оказалось, что более подходящими являются увеличенный параметр $\alpha = 10^{-2}$ и параметр $\mu = 1$. На рис. 3 приведено точное решение второй задачи, описание решения которой изложено выше. На рис. 4 приведено приближенное решение этой задачи со значениями параметров, выбранными в процессе отбора по результатам просмотра получающихся приближений, $\alpha = 10^{-11}$, $\mu = 0.01$, $H = 1$. Процесс промежуточных расчетов сопровождается выводом не только промежуточных решений, но и промежуточных градиентов целевого функционала, рис. 6 и 7, участвующих в расчетных формулах типа (4).

Следует отметить, что одновременно с графиками выдаются соответствующие числовые данные, поэтому отпадает необходимость сохранения и упорядочения сохраняемых промежуточных данных для последующего анализа прохождения счета, оценки и принятия решения об изменении параметров счета.

Временные издержки, возникающие при применении подсистемы онлайн-визуализации, во многих случаях оказываются существенно ниже, чем при использовании внешних графических программ, когда помимо расходов на структуризацию достаточно большого объема результатов счета время тратится и на действия по переходу от одной системы к другой. Так, в случае первой из рассматриваемых задач — восстановлении двумерного изображения — приходилось останавливать счет и переходить к использованию графических возможностей системы MatLab.

Применение онлайн-визуализации для второй задачи, благодаря достаточно короткому времени отклика, способствовало проведению дополнительных численных экспериментов за счет освобождения времени у исследователя. А, увидев, что у текущих графиков отсутствует ожидаемая симметрия, удалось найти и исправить возникшую опечатку в тексте программы. Кроме того, в результате экспериментов с изменением параметра $H = 0.05$ (входящего в подинтегральное ядро), получили приближение, график которого практически совпадает с графиком точного решения, изображенного на рис. 3.

4. Заключение

По сути, система онлайн-визуализации является инструментом, позволяющим использовать при интерпретации результатов счета и аналитический подход, и зрительное восприятие, ориентированное на быстрый анализ ситуации в целом. В результате применения данной системы сокращается время интерпретации результатов счета и интервал между вычислительными экспериментами. Пользователь получает новые возможности для анализа корректности и производительности вычислительной программы, что способствует получению качественных выводов и выбору подходящих параметров, а в конечном итоге, построению хороших приближений к точному решению.

В заключение еще хотелось бы подчеркнуть полезность взаимодействия квалифицированных математиков, прикладных и системных программистов. В результате такого взаимодействия, в данном случае, в процессе численных расчетов были опробованы возможности вычислительных методов, разработанных в теоретическом плане и, с другой стороны, разработаны и эффективно применены новые технические и методологические средства проведения вычислительного эксперимента, которые могут быть адаптированы для решения и других задач.

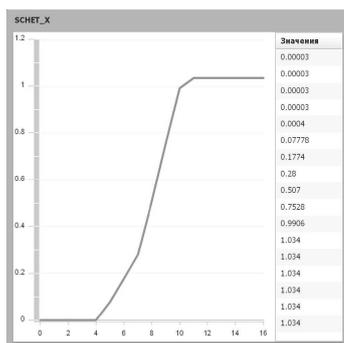


Рис. 1

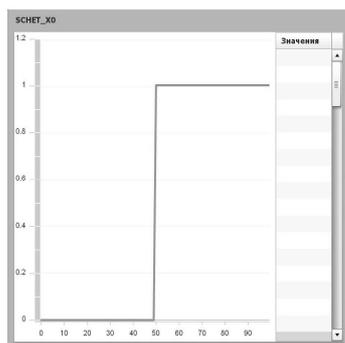


Рис. 2

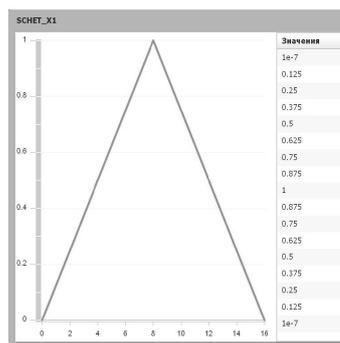


Рис. 3

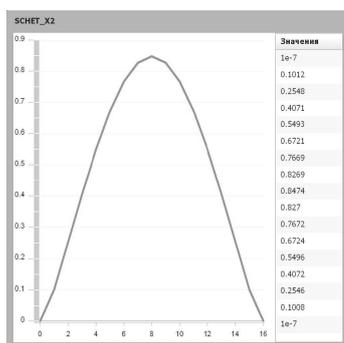


Рис. 4

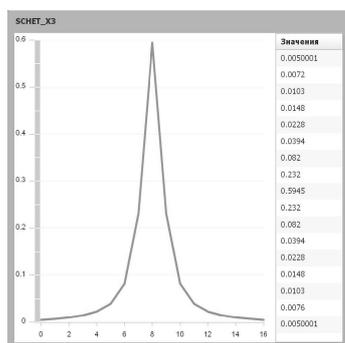


Рис. 5

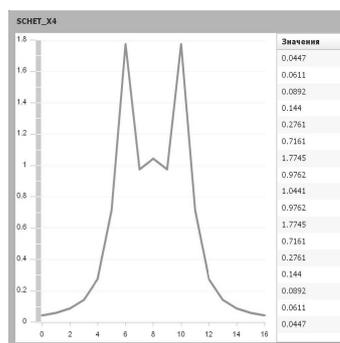


Рис. 6

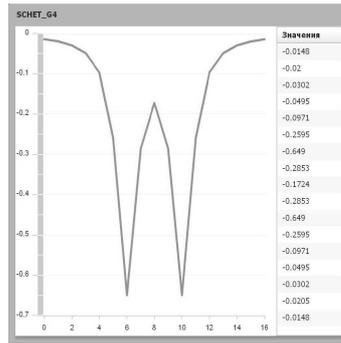


Рис. 7

шЄхЁрЄєЁр

1. Васин В.В. Регуляризация и итеративная аппроксимация для линейных некорректных задач в пространстве функций ограниченной вариации // Труды ИММ УрО РАН. 2002. Т.8, №1. С.189-202.
2. Васин В.В. Аппроксимация негладких решений линейных некорректных задач // Труды ИММ УрО РАН. 2006. Т.12, №1. С.64-77.
3. Васин В.В., Серезникова Т.И., Шарф С.В. Решение одной задачи восстановления изображения методом регуляризации и итеративной аппроксимации // Алгоритмы и программные средства параллельных вычислений. Вып.7. Екатеринбург: УрО РАН. 2001. С.250-261