

УДК 519.6

ОСОБЕННОСТИ РЕШЕНИЯ ЗАДАЧ КАРТИРОВАНИЯ РАСПРЕДЕЛЕНИЯ ХИМИЧЕСКИХ ЭЛЕМЕНТОВ ПО ПОВЕРХНОСТЯМ ЗВЕЗД КАК НЕКОРРЕКТНЫХ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ МНОГОПРОЦЕССОРНЫХ СИСТЕМ

А. Г. Ягола¹, В. Н. Титаренко¹, М. П. Васильев¹, Е. В. Шимановская¹

Рассматривается решение задачи картирования распределения химических элементов по поверхностям звезд как некорректной задачи. Минимизация функционала Тихонова с выбором параметра регуляризации по конечномерному обобщенному принципу невязки используется для построения регуляризирующего алгоритма решения рассматриваемой задачи. В качестве метода минимизации применяется метод проекции сопряженных градиентов на множество неотрицательных векторов. Рассматриваются некоторые особенности численного решения задачи минимизации для задачи картирования. Для численного решения модельных задач используется многопроцессорный компьютер. Предлагаются схемы распараллеливания исходной задачи, обсуждаются особенности их реализации.

Ключевые слова: некорректная задача, задача картирования, функционал Тихонова, многопроцессорные системы, доплеровское смещение.

1. Введение. Для многих звезд характерно периодическое изменение профилей линий поглощения некоторых химических элементов. Такие наблюдаемые изменения профилей объясняются неравномерностью распределения химических элементов по поверхностям вращающихся звезд и как следствие — различием профилей линий поглощения на различных участках звездных поверхностей. Спектральный состав непрерывного излучения и блеск звезды, определяемые в основном физическими условиями в атмосфере, не изменяются или изменяются слабо. Поэтому переменность профилей обуславливается именно неоднородностью химического состава, а не различием физических условий на поверхности. Подтверждением данного утверждения является тот факт, что профили разных линий одного химического элемента, соответствующие различным степеням его ионизации, изменяются синхронно, а профили различных элементов с близкими потенциалами возбуждения и ионизации изменяются асинхронно.

В настоящее время в астрофизике очень часто возникают задачи восстановления распределения магнитного поля на различных звездах. Описанные же выше звезды почти всегда обладают значительными магнитными полями, изменяющимися с теми же периодами, что и профили линий. Поэтому велика вероятность того, что неоднородность химического состава обусловлена наличием на звезде сильного магнитного поля. В связи с этим очень интересно получить распределение химического состава звезды для последующего сравнения с распределением магнитного поля.

2. Постановка задачи. Рассмотрим вопрос о профилях линий поглощения в спектре звезды, наблюдаемом с Земли. Постановку задачи производим как и в [1, 2]. По определению, количество энергии dE , переносимое излучением через единичную площадку поверхности под углом θ к ее нормали, в телесном угле $2\pi \sin \theta d\theta$ в интервале длин волн $[\lambda, \lambda + d\lambda]$ и в интервале времени $[t, t + dt]$ равно

$$dE = 2\pi I(M, \theta, \lambda) \sin \theta d\theta \cos \theta d\lambda dt, \tag{2.1}$$

где $I(M, \theta, \lambda)$ — интенсивность излучения в рассматриваемой точке M поверхности звезды. Энергия, входящая на Землю от всей видимой поверхности звезды, пропорциональна $F_{\lambda}^+ = \iint I(M, \theta, \lambda) \cos \theta dM$, где dM — элемент сферической поверхности, θ — угол между нормалью к поверхности звезды в точке M и лучом зрения, интегрирование производится по всей видимой поверхности звезды. В качестве системы координат на данном этапе введем сферическую систему координат (θ, φ) с осью, направленной по лучу зрения (см. рис. 1). Тогда элемент поверхности равен $dM = \sin \theta d\theta d\varphi$, а видимая поверхность определяется условием $\cos \theta \geq 0$.

Следует учесть, что для быстро вращающихся звезд доплеровское смещение излучения в различных точках поверхности звезды различно и может достигать значений, сравнимых с шириной линии. Поэтому

¹ Московский государственный университет им. М. В. Ломоносова, физический факультет, Воробьевы горы, 119992, Москва; e-mail: yagola@inverse.phys.msu.su; ill-posed@mail.ru

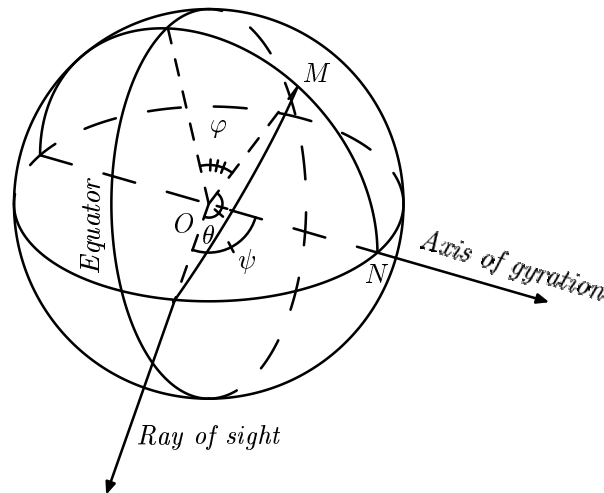


Рис. 1. Система координат на вращающейся звезде

в интеграле для F_{λ}^{+} следует изменить аргумент λ в функции $I(M, \theta, \lambda)$ на $\lambda + \Delta\lambda_D(M)$, где $\Delta\lambda_D(M)$ описывает величину доплеровского смещения излучения.

Следует различать интенсивность излучения в непрерывном спектре $I_0(M, \theta, \lambda)$ и интенсивность излучения в линии. Поскольку ширины рассматриваемых линий обычно малы, то в пределах линии интенсивность излучения в непрерывном спектре практически не изменяется. Предположим, что интенсивность в непрерывном спектре не зависит от точки M на поверхности звезды, так как физические условия на звезде приблизительно однородны. В соответствии со сделанными замечаниями $I_0(M, \theta, \lambda) = I_0(\theta)$.

Зависимость интенсивности излучения в непрерывном спектре от угла (так называемое потемнение к краю) будем рассматривать в виде $I_0(\theta) = I_0 u_1(\theta)$, где $u_1(\theta)$ — линейная функция $\cos \theta$. Принимаем, что $u_1(\theta) = 1 - u_1 + u_1 \cos \theta$. Зависимость коэффициента u_1 от температуры хорошо изучена, а температура звезды может быть определена по ее спектральному классу.

Исследования показали, что функция

$$r(M, \lambda) = 1 - \frac{I(M, 0, \lambda)}{I_0(M, 0)} \quad (2.2)$$

во всех случаях хорошо представляется формулой Миннарта

$$r(M, \lambda) = \left(\frac{1}{x(M, \lambda)} + \frac{1}{R_c(M)} \right)^{-1}, \quad (2.3)$$

где $R_c(M)$ — центральная глубина линии, $x(M, \lambda) = z(M)k(\lambda)$, а $k(\lambda)$ — так называемый фойгтовский профиль. Коэффициент $z(M)$ определяется относительным содержанием поглощающего элемента в атмосфере.

Зависимость интенсивности излучения в линии $I(M, \theta, \lambda)$ от угла θ может быть очень сложной. Однако в статье будем предполагать, что можно записать

$$1 - \frac{I(M, \theta, \lambda)}{I_0(M, \theta)} = r(M, \lambda) u_2(\theta), \quad (2.4)$$

где функция $u_2(\theta)$ линейна относительно $\cos \theta$: $u_2(\theta) = 1 - u_2 + u_2 \cos \theta$. Параметр u_2 находится путем численного интегрирования уравнений переноса для атмосферы звезды. Часто u_2 принимается равным нулю.

В условиях однородности физических условий параметры R_c и z формулы Миннарта (2.3) не являются независимыми. Величина R_c определяет центральную глубину линии, а z — ее ширину. Известно, что зависимость центральной глубины от величины z можно аппроксимировать формулой

$$R_c = h(z) = C_1 (1 - e^{-C_2 z}), \quad (2.5)$$

где параметры C_1, C_2 выбираются на основании исследования модели атмосферы звезды.

В окончательном виде наблюдаемые на Земле профили линий поглощения в спектре звезды определяются двумерным интегральным уравнением, нелинейным по $z(M)$:

$$r(\lambda) = \frac{1}{A} \iint_{\cos \theta \geq 0} f(z(M), \lambda + \Delta\lambda_D(M)) u_1(\theta) u_2(\theta) \cos \theta dM, \quad (2.6)$$

где

$$A = 2\pi \int_0^{\pi/2} \cos \theta u_1(\theta) \sin \theta d\theta, \quad f(z, \lambda) = \left(\frac{1}{h(z)} + \frac{1}{zk(\lambda)} \right)^{-1}. \quad (2.7)$$

При вращении звезды профили $R(\lambda)$ изменяются. Поэтому, если t — время, а ω — угловая скорость вращения звезды, то интегральное уравнение (2.6) переписывается в виде

$$R(\lambda, \omega t) = \iint f(z(M), \lambda + \Delta\lambda_D(M, \omega t)) H(M, \omega t) dM. \quad (2.8)$$

В качестве координат на поверхности звезды выбираем широту φ и долготу ψ (см. рис. 1). При этом элемент поверхности $dM = \cos \varphi d\varphi d\psi$,

$$\Delta\lambda_D(\varphi, \psi, \omega t) = \frac{\lambda_0}{c} v_{eq} \sin \beta \cos \varphi \cos(\psi - \omega t), \quad (2.9)$$

$$H(\varphi, \psi, \omega t) = \begin{cases} \frac{1}{A} u_1(\theta) u_2(\theta) \cos \theta \cos \varphi, & \cos \theta > 0, \\ 0, & \cos \theta \leq 0, \end{cases} \quad (2.10)$$

$$\cos \theta = \cos \beta \sin \varphi + \cos \varphi \sin \beta \cos(\psi - \omega t), \quad (2.11)$$

где v_{eq} — экваториальная скорость вращения звезды, а β — угол наклона оси вращения звезды к лучу зрения.

Фойгтовский профиль $k(\lambda)$, определяющий зависимость профиля линии от длины волны, является сверткой двух функций, описывающих профиль линии, уширенной за счет давления (столкновений) и доплеровского смещения при микротурбулентных движениях в атмосфере звезды:

$$k(\lambda) = \frac{\varkappa_l}{\pi \varkappa_d} \int_{-\infty}^{+\infty} \frac{e^{-t^2} dt}{\frac{\varkappa_l^2}{\varkappa_d^2} + \left(t - \frac{\lambda}{\varkappa_d} \right)^2}. \quad (2.12)$$

Здесь \varkappa_l — величина лоренцевского уширения давлением, а параметр \varkappa_d характеризует величину уширения линии микротурбулентными движениями.

Ставим своей задачей нахождение функции $z(M)$ по измеренным в различные фазы профилям линии — функциям $R(\lambda, \omega t)$. На функцию $z(M)$ должно быть наложено естественное ограничение $z(M) \geq 0$. Постановка задачи в такой форме впервые была дана в работе [3].

3. Метод регуляризации. Нелинейное интегральное уравнение (2.8) является уравнением Фредгольма первого рода, т.е. некорректной задачей; поэтому для его решения необходимо строить так называемые регуляризирующие алгоритмы. Воспользуемся алгоритмами, основанными на минимизации сглаживающего функционала Тихонова. Для уравнения (2.8) запишем функционал Тихонова:

$$M^\alpha[z] = I[z] + \alpha \Omega[z] \equiv \iint d\lambda d(\omega t) \left\{ R(\lambda, \omega t) - \iint f(z(M), \lambda + \Delta\lambda_D(M, \omega t)) H(M, \omega t) dM \right\}^2 + \alpha \iint \left\{ z^2 + \left(\frac{\partial z}{\partial \varphi} \right)^2 + \left(\frac{\partial z}{\partial \psi} \right)^2 \right\} d\varphi d\psi. \quad (3.1)$$

Предполагаем, что функция $z(M)$ принадлежит пространству W_2^1 на поверхности звезды, причем минимизацию функционала необходимо производить на множестве функций $z(M) \geq 0$. В качестве области определения функции $z(M)$ берем множество $(\varphi, \psi) \in [-\beta, \frac{\pi}{2}] \times [0, 2\pi]$, так как участки звезды с $\varphi \in [-\frac{\pi}{2}, -\beta]$ не видны с Земли. Функцию $R(\lambda, \omega t)$, описывающую экспериментальные данные, считаем заданной при $\lambda \in \left[\lambda_0 - \frac{\Delta\lambda}{2}, \lambda_0 + \frac{\Delta\lambda}{2} \right]$ и $t \in [0, T]$.

Для численного решения задачи необходимо перейти к конечномерным пространствам. Для этого введем на поверхности звезды прямоугольную сетку $\{\varphi_i, \psi_j\}$, представляющую собой декартово произведение сеток $\{\varphi_i\}_1^n, \{\psi_j\}_1^m$, таких, что $-\beta \leq \varphi_1 < \varphi_2 < \dots < \varphi_n \leq \frac{\pi}{2}$, $0 \leq \psi_1 < \psi_2 < \dots < \psi_m \leq 2\pi$. Считаем, что сетки $\{\varphi_i\}_1^n, \{\psi_j\}_1^m$ равномерные и задаются формулами $\varphi_i = -\beta + \frac{h_\varphi}{2} + h_\varphi(i-1)$, $i = \overline{1, n}$; $\psi_j = h_\psi(j-1)$, $j = \overline{1, m}$, где шаги $h_\varphi = \frac{\pi/2 + \beta}{n}$, $h_\psi = \frac{2\pi}{m}$. Примем, что $z_{ij} = z(\varphi_i, \psi_j)$, а \hat{z} — вектор z_{ij} , $i = \overline{1, n}$, $j = \overline{1, m}$. По длине волны λ и по фазе ωt введем равномерные сетки $\{\lambda_k\}_1^{N_\lambda}$ и $\{(\omega t)_l\}_1^{N_{\omega t}}$, такие, что $h_\lambda = \frac{\Delta\lambda}{N_\lambda - 1}$, $\lambda_k = \lambda_0 - \frac{\Delta\lambda}{2} + h_\lambda(k-1)$, $k = \overline{1, N_\lambda}$; $h_{\omega t} = \frac{\omega T}{N_{\omega t} - 1}$, $(\omega t)_l = h_{\omega t}(l-1)$, $l = \overline{1, N_{\omega t}}$. Все интегралы в формуле (3.1) аппроксимируются по формуле прямоугольников. Таким образом, получаем конечноразностную аппроксимацию функционала:

$$\widehat{M}^\alpha[\hat{z}] = \hat{I}[\hat{z}] + \alpha \widehat{\Omega}[\hat{z}], \quad (3.2)$$

где $\hat{I}[\hat{z}] \equiv g(\hat{J}[\hat{z}])$, $g(x) \equiv x$ и

$$\hat{J}[\hat{z}] = \sum_{k=1}^{N_\lambda} \sum_{l=1}^{N_{\omega t}} W_{kl} \left\{ R_{kl} - \sum_{i=1}^n \sum_{j=1}^m h_\varphi h_\psi H(\varphi_i, \psi_j, (\omega t)_l) f(z_{ij}, \lambda_k + \Delta\lambda_D(\varphi_i, \psi_j, (\omega t)_l)) \right\}^2, \quad (3.3)$$

$$\widehat{\Omega}[\hat{z}] = h_\varphi h_\psi \sum_{i=1}^n \sum_{j=1}^m z_{ij}^2 + \frac{h_\psi}{h_\varphi} \sum_{i=2}^n \sum_{j=1}^m (z_{ij} - z_{i-1,j})^2 + \frac{h_\varphi}{h_\psi} \sum_{i=1}^n \left\{ \sum_{j=2}^m (z_{ij} - z_{i,j-1})^2 + (z_{i1} - z_{im})^2 \right\}. \quad (3.4)$$

Здесь $W_{kl} > 0$ — веса, введение которых мотивируется тем, что точность измерений $R(\lambda, \omega t)$ и число измерений $R(\lambda, \omega t)$ в разных фазах различно; профили, соответствующие каждой фазе, обычно известны с большей точностью в центре линии и с меньшей — на ее краях. При этом для согласования с бесконечномерной задачей считаем, что $W_{kl} \sim h_\lambda h_{\omega t}$ (во всех модельных задачах, рассматриваемых ниже, просто принимаем, что $W_{kl} = h_\lambda h_{\omega t}$). Таким образом, исходная задача сводится к задаче минимизации в N -мерном пространстве, где $N \equiv n \times m$, на множестве \widehat{D} неотрицательных векторов z_{ij} с последующим выбором параметра регуляризации α по некоторому правилу.

Выбор функционалов $\widehat{\Omega}[\hat{z}]$, $\hat{J}[\hat{z}]$, $g(x)$ в рассматриваемом виде обеспечивает выполнение всех условий (см. [2]) для построения регуляризирующих алгоритмов для решаемой нелинейной некорректной задачи.

4. Выбор параметра регуляризации. Считаем, что погрешность δ входных данных связана только с погрешностью задания профилей линий поглощения звезды. Принимаем, что

$$\delta^2 \equiv \iint d\lambda d(\omega t) (R(\lambda, \omega t) - R_\delta(\lambda, \omega t))^2, \quad (4.1)$$

где $R(\lambda, \omega t)$ — точный профиль линии, а $R_\delta(\lambda, \omega t)$ — измеренный в эксперименте. Тогда вместо точного функционала $J[z]$ имеется приближенный функционал $J_\delta[z]$, для которого выполняется условие аппроксимации

$$|J[z] - J_\delta[z]| \leq \Psi_0(\delta, \Omega[z]). \quad (4.2)$$

Вместо исходного функционала $J_\delta[z]$ имеется его конечномерная аппроксимация $\hat{J}_\delta[\hat{z}]$, удовлетворяющая следующему условию аппроксимации:

$$|\hat{J}_\delta[\hat{z}] - J_\delta[z]| \leq \widehat{\Psi}(\eta, \Omega[z]). \quad (4.3)$$

В качестве вектора погрешности η рассматриваем вектор $\eta \equiv (\delta, 1/N, 1/\tilde{N})$, где $N = n \times m$, $\tilde{N} = N_\lambda \times N_{\omega t}$. Ясно, что $\eta \rightarrow 0$ при $\delta \rightarrow 0$, $N, \tilde{N} \rightarrow \infty$.

Для построения регуляризирующего алгоритма меры аппроксимации $\Psi_0(\delta, \Omega[z])$ и $\widehat{\Psi}(\eta, \Omega[z])$ должны удовлетворять некоторым условиям [2], которые в статье предполагаются выполненными. Из условий аппроксимации (4.2) и (4.3) следует, что выполняется неравенство

$$|J[z] - \hat{J}_\eta[\hat{z}]| \leq \Psi_0(\delta, \Omega[z]) + \widehat{\Psi}(\eta, \Omega[z]) \equiv \Psi(\eta, \Omega[z]). \quad (4.4)$$

Функция $\Psi(\eta, \Omega[z])$ представляет собой совокупную меру аппроксимации точного функционала $J[z]$ конечномерным функционалом $\hat{J}_\eta[\hat{z}]$. На рис. 2 и рис. 3 показаны зависимости функции $\Psi(\eta, \Omega[z])$ от размерностей задачи N и \tilde{N} , погрешности δ и значения стабилизирующего функционала $\Omega[z]$. Для построения этих зависимостей была рассмотрена модельная задача.

Задача 1: $T = 10^5$, $\beta = 40^\circ$, $\omega = 10^{-4}$, $v_{eq} = 10^5$, $\lambda_0 = 10^{-6}$, $\Delta\lambda = 10^{-9}$, $C_1 = 1$, $C_2 = 10^{-3}$, $u_1 = 0.5$, $u_2 = 0.5$, $\varkappa_d = 1$, $\varkappa_l = 10^{-3}$, $c = 3 \cdot 10^8$.

Интегральное уравнение (3.1) нелинейно зависит от функции $z(M)$. Из этого следует, что аналитически найти меру аппроксимации $\Psi(\eta, \Omega[z])$ очень сложно. Поэтому для ее нахождения рассматривались различные модельные функции $z(M)$ и в качестве значения функции $\Psi(\eta, \Omega[z])$ выбиралось максимальное значение для рассматриваемых функций при фиксированных значениях η и $\Omega[z]$.

На рис. 2 а показано, что при относительно небольших размерностях N вектора \hat{z} ($N \approx 600$) наступает насыщение меры аппроксимации $\Psi(\eta, \Omega[z])$. Зависимость $\Psi(\eta, \Omega[z])$ от размерности \tilde{N} вектора входных данных, изображенная на рис. 2 б, очень четко прослеживается: $\Psi(\eta, \Omega[z]) \sim \tilde{N}^{-2}$.

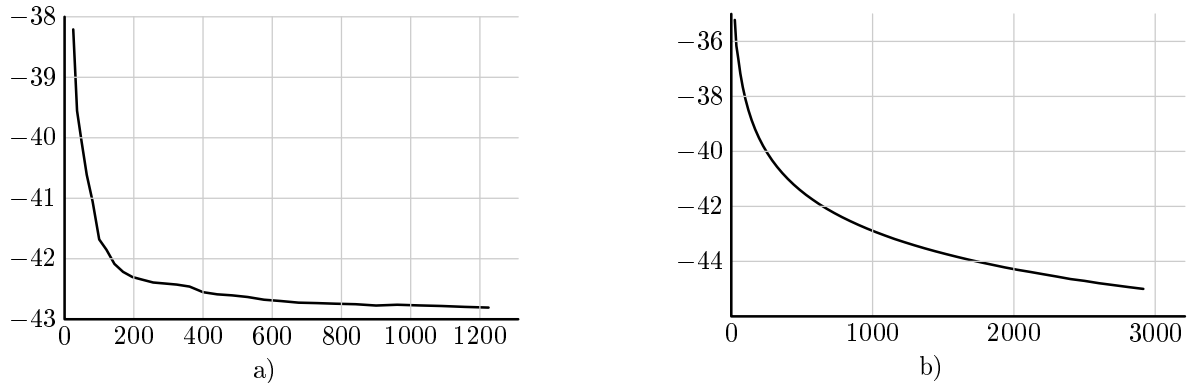


Рис. 2. а) Зависимость $\ln(\Psi(\eta, \Omega[z]))$ от N при $\tilde{N} = 900$, $\Omega[z] = 1$, $\delta = 0$. б) Зависимость $\ln(\Psi(\eta, \Omega[z]))$ от \tilde{N} при $N = 900$, $\Omega[z] = 1$, $\delta = 0$

Зависимость $\ln(\Psi(\eta, \Omega[z]))$ от натурального логарифма $\ln \delta$ погрешности задания профилей линий поглощения изображена на рис. 3 а. Для рассматриваемой зависимости характерно насыщение при малых значениях δ , т.е. существует горизонтальная асимптота для $\Psi(\eta, \Omega[z])$ при $\delta \rightarrow 0$. При больших δ зависимость меры аппроксимации $\Psi(\eta, \Omega[z])$ становится линейной по δ , а именно $\Psi(\eta, \Omega[z]) \approx \delta$.

На рис. 3 б изображена зависимость $\ln(\Psi(\eta, \Omega[z]))$ от $\ln(\Omega[z])$ при $\delta = 0$. Эта зависимость линейна по $\ln(\Omega[z])$. При $\delta \neq 0$ вид функции $\ln(\Psi(\eta, \Omega[z]))$ от $\ln(\Omega[z])$ становится похожим на зависимость $\ln(\Psi(\eta, \Omega[z]))$ от $\ln \delta$, т.е. существуют горизонтальная и наклонная асимптоты. При этом угол наклона для наклонной асимптоты тем больше, чем меньше значение δ .

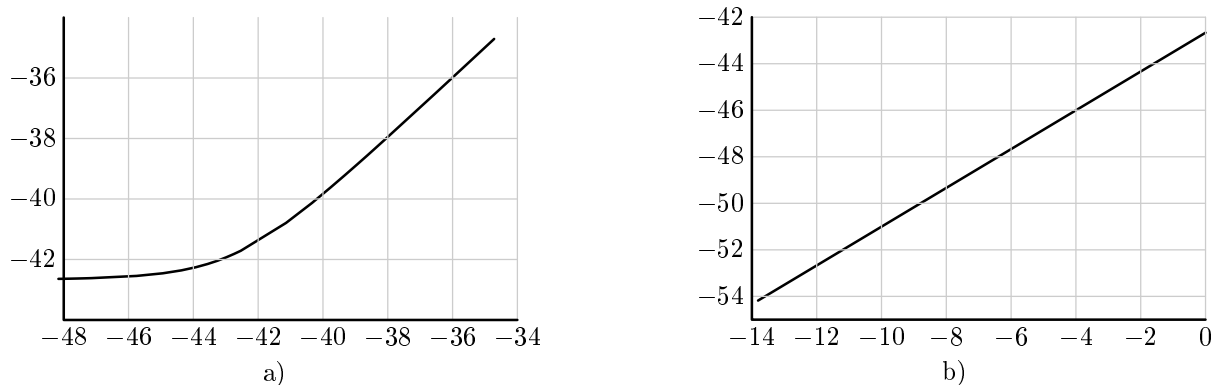


Рис. 3. а) Зависимость $\ln(\Psi(\eta, \Omega[z]))$ от $\ln \delta$ при $N = \tilde{N} = 900$, $\Omega[z] = 1$. б) Зависимость $\ln(\Psi(\eta, \Omega[z]))$ от $\ln(\Omega[z])$ при $N = \tilde{N} = 900$, $\delta = 0$

При решении некорректной задачи очень важен выбор параметра регуляризации α . Параметр регуляризации должен зависеть от входных данных, их погрешностей и способа аппроксимации исходной задачи. В качестве алгоритмов для нахождения параметра регуляризации α можно использовать конечномерные обобщенные принципы невязки, квазирешений, сглаживающего функционала, описание которых можно найти в [2].

Определим при фиксированном η величины:

$$\hat{\lambda} \equiv \hat{\lambda}_\eta = \inf \left\{ \hat{J}_\eta[\hat{z}] + \Psi(\eta, \hat{\Omega}[\hat{z}]) : \hat{z} \in \hat{D} \right\}, \quad (4.5)$$

$$\hat{\pi}(\alpha) = f(\hat{\lambda} + \Psi(\eta, \hat{\Omega}[\hat{z}^\alpha])), \quad (4.6)$$

$$\hat{\rho}(\alpha) = \hat{I}[\hat{z}^\alpha] - \hat{\pi}(\alpha), \quad (4.7)$$

где $\hat{z}^\alpha \in \hat{Z}^\alpha \equiv \text{Arg inf} \{ \widehat{M}^\alpha[\hat{z}] : \hat{z} \in \hat{D} \}$. Выбор параметра регуляризации α по алгоритму конечномерного обобщенного принципа невязки заключается в нахождении решения уравнения с монотонной функцией

$$\hat{\rho}(\alpha) = 0. \quad (4.8)$$

Тогда $z \equiv \bar{P}\hat{z} \rightarrow \bar{z}$ при $\eta \rightarrow 0$, где $\bar{P} : \hat{z} \rightarrow z$.

5. Минимизация сглаживающего функционала. Для минимизации сглаживающего функционала $\widehat{M}^\alpha[\hat{z}]$ используем метод проекции сопряженных градиентов. Итерационный процесс строится следующим образом: задается некоторое начальное приближение — вектор $\hat{z}^{(0)}$ с неотрицательными компонентами. Для заданной точки $\hat{z}^{(i)}$ минимизирующей последовательности вычисляется градиент $g^{(i)} = \text{grad} \widehat{M}^\alpha[\hat{z}^{(i)}] = \text{grad} \hat{I}[\hat{z}^{(i)}] + \alpha \text{grad} \hat{\Omega}[\hat{z}^{(i)}]$, где для нахождения градиентов функционалов $\hat{I}[\hat{z}]$, $\hat{\Omega}[\hat{z}]$ используются аналитические выражения, которые легко получить из формул (3.3), (3.4):

$$\frac{\partial \hat{I}}{\partial z_{ij}} = -2h_\varphi h_\psi \sum_{l=1}^{N_{\omega t}} H(\varphi_i, \psi_j, (\omega t)_l) \sum_{k=1}^{N_\lambda} \left[W_{kl} \left\{ \dots \right\} \frac{\partial f}{\partial z_{ij}}(z_{ij}, \lambda_k + \Delta \lambda_D(\varphi_i, \psi_j, (\omega t)_l)) \right], \quad (5.1)$$

$$\frac{\partial \hat{\Omega}}{\partial z_{ij}} = 2h_\varphi h_\psi z_{ij} + 2 \frac{h_\psi}{h_\varphi} \Omega'_1 + 2 \frac{h_\varphi}{h_\psi} \Omega'_2, \quad (5.2)$$

$$\Omega'_1 = 2z_{ij} - \begin{cases} z_{ij} + z_{i+1j}, & i = 1, \\ z_{i-1j} + z_{i+1j}, & i \in \overline{2, n-1}, \\ z_{i-1j} + z_{ij}, & i = n, \end{cases} \quad \Omega'_2 = 2z_{ij} - \begin{cases} z_{ij+1} + z_{im}, & j = 1, \\ z_{ij-1} + z_{ij+1}, & j \in \overline{2, m-1}, \\ z_{i1} + z_{ij-1}, & j = m. \end{cases} \quad (5.3)$$

Как видно из этих формул, для вычисления $\text{grad} \hat{I}[\hat{z}]$ необходимо вычислить $\frac{\partial}{\partial z} f(z, \lambda)$:

$$\frac{\partial}{\partial z} f(z, \lambda) = k(\lambda) \frac{z^2 k(\lambda) h'(z) + h^2(z)}{(zk(\lambda) + h(z))^2}, \quad \frac{\partial}{\partial z} h(z) = C_1 C_2 e^{-C_2 z}. \quad (5.4)$$

Отметим, что функции $f(z, \lambda)$, $\frac{\partial}{\partial z} f(z, \lambda)$ не определены в точке $z = 0$, но являются непрерывными справа. Поэтому доопределим их в рассматриваемой точке по непрерывности:

$$f(0, \lambda) = 0, \quad \frac{\partial}{\partial z} f(0, \lambda) = \frac{C_1 C_2 k(\lambda)}{k(\lambda) + C_1 C_2}. \quad (5.5)$$

По направлению градиента $g^{(i-1)}$ и направлению спуска $h^{(i-1)}$ на предыдущем шаге вычисляется новое направление спуска по формуле

$$h^{(i)} = g^{(i)} + \gamma_i h^{(i-1)}, \quad \gamma_i = \frac{(g^{(i)} - g^{(i-1)}, g^{(i)})}{(g^{(i-1)}, g^{(i-1)})} \quad (5.6)$$

(вариант Полака–Рибьера метода сопряженных градиентов). После этого строится однопараметрическое множество, состоящее из элементов

$$\hat{z}_\lambda = P(\hat{z}^{(i)} - \lambda h^{(i)}), \quad \lambda \geq 0, \quad (5.7)$$

где P — оператор проектирования N -мерного вектора на неотрицательный октант \hat{D} . Затем решается задача одномерной минимизации $\widehat{M}^\alpha[\hat{z}]$ на множестве \hat{D} . Точка минимума функционала $\widehat{M}^\alpha[\hat{z}]$ на рассматриваемом множестве \hat{D} принимается за следующий элемент $\hat{z}^{(i+1)}$ минимизирующей последовательности. При решении одномерной задачи минимизации используется квадратичная аппроксимация функции

$$\sigma(\lambda) \equiv \widehat{M}^\alpha [P(\hat{z}^{(i)} - \lambda h^{(i)})] - \widehat{M}^\alpha [\hat{z}^{(i)}] \quad (5.8)$$

по трем точкам 0 , λ_{step} , $2\lambda_{\text{step}}$. После этого приближенное решение одномерной задачи минимизации берется в виде

$$\lambda_{\min} = \lambda_{\text{step}} \cdot \frac{1}{2} \frac{\sigma_2 - 4\sigma_1}{\sigma_2 - 2\sigma_1}, \quad \sigma_1 \equiv \sigma(\lambda_{\text{step}}), \quad \sigma_2 \equiv \sigma(2\lambda_{\text{step}}). \quad (5.9)$$

При необходимости производится дополнительное уточнение этой точки. Через каждые N шагов минимизации производится “обновление” метода, т.е. γ_i полагается равным нулю в формуле (5.6). Кроме того, алгоритм обновляется, если вследствие неточности решения одномерной задачи минимизации на очередном шаге получится $(h^{(i)}, g^{(i)}) \leq 0$.

6. Некоторые особенности численной реализации. Рассмотрим некоторые особенности численного решения поставленной выше задачи. Согласно формуле (2.12), фойгтовский профиль $k(\lambda)$ является сверткой двух функций. Вычисление этой свертки является трудоемкой операцией, проводить которую каждый раз не рекомендуется. Поэтому можно поступить следующим образом: аппроксимировать функцию $k(\lambda)$ линейной интерполяцией по сеточным значениям на некоторой сетке. Можно также вычислять функцию $k(\lambda)$, используя ее аппроксимацию по формуле

$$k(\lambda) = \frac{\sqrt{\ln 2}}{\sqrt{\pi} \kappa_v} (1 - \xi) 2^{-\eta^2} + \frac{1}{\pi \kappa_v} \frac{1}{1 + \eta^2} - \frac{\xi(1 - \xi)}{\pi \kappa_v} \left(\frac{3}{2 \ln 2} + 1 + \xi \right) \left[0.066 e^{-0.4 \eta^2} - \frac{1}{40 - 5.5 \eta^2 + \eta^4} \right], \quad (6.1)$$

$$\kappa_v = \frac{1}{2} \left(\kappa_l + \sqrt{\kappa_l^2 + 4 \kappa_d^2} \right) + 0.05 \kappa_l \left(1 - \frac{2 \kappa_l}{\kappa_l + \sqrt{\kappa_l^2 + \kappa_d^2}} \right), \quad \eta = \frac{\lambda - \lambda_0}{\kappa_v}, \quad \xi = \frac{\kappa_l}{\kappa_v}. \quad (6.2)$$

Из формул (3.3) и (5.1) для функционала невязки $\hat{I}[\hat{z}]$ и его градиента $\frac{\partial \hat{I}}{\partial z_{ij}}$ видно, что при суммировании часто используются функции, заданные на одномерных, двумерных или трехмерных сетках, например, функция $H(\varphi_i, \psi_j, (\omega t)_l)$. Такие функции можно предварительно вычислить, т.е. построить матрицу H_{ijl} , а потом уже использовать при вычислениях. Другие вычисления (например, $\psi - \omega t$ для нахождения доплеровского смещения $\Delta \lambda_D(\varphi_i, \psi_j, (\omega t)_l)$ или $\cos \varphi$) можно “вынести за сумму”, т.е. организовать суммирование так, чтобы вычисление некоторого выражения, общего для некоторой суммы (в данном примере для суммы по k в формулах (3.3), (5.1)) происходило до вычисления этой суммы, а потом при суммировании полученный результат просто подставлялся. Ясно, что при этом усложняется программа для решения поставленной задачи. Но выигрыш по времени работы очень существенен (в разы). Это связано с тем, что все рассматриваемые в задаче функции имеют простую структуру, а именно, представляют собой суммы легко вычисляемых на компьютере выражений. Но суммирование может осуществляться по шести переменным, как для $\frac{\partial \hat{I}}{\partial z_{ij}}$, что и приводит к большим временам вычислений. Поэтому правильная организация порядка суммирования и последовательности вычислений существенно уменьшает время работы программы.

При минимизации выпуклой функции на некотором множестве с использованием методов первого порядка, требующих вычисления значения функции и ее производной в точке, всегда возникает следующий вопрос. Насколько точно необходимо решать задачу одномерной минимизации? Ответ на этот вопрос зависит от вида минимизируемой функции. Если время вычисления градиента функции не слишком превышает время вычисления значения функции в точке, то задачу одномерной минимизации можно решать не очень точно. Это связано с тем, что за время, потраченное на точное нахождение точки минимума одномерной функции, можно совершить несколько итераций для многомерной минимизации. При этом находится точка, в которой значение функции часто оказывается меньше, чем минимальное значение функции на рассматриваемом выше одномерном множестве. Если же вычисление градиента требует много времени, то одномерная минимизация должна быть проведена более точно. Для рассматриваемого в данной работе функционала $\widehat{M}^\alpha[\hat{z}]$ на рис. 4 приведены отношения времен вычисления градиента функционала и самого функционала в зависимости от размерностей N и \tilde{N} вектора \hat{z} и входных данных соответственно. Видно, что в первом случае зависимость является линейной по N , а во втором случае можно считать, что нет зависимости от \tilde{N} . Из полученных соотношений для времен вычисления функционала и его градиента следует, что для рассматриваемой задачи многомерной минимизации задача одномерной минимизации должна решаться как можно точнее.

Важен вопрос выбора шага λ_{step} для одномерной минимизации выпуклой функции $\sigma(\lambda)$. Если выбрать λ_{step} достаточно маленьким, то может оказаться, что значения σ_1, σ_2 близки и при их вычислении начинают влиять погрешности округления. Таким образом, вычисленные значения могут и не являться значениями выпуклой вниз функции. Если же шаг λ_{step} слишком большой, то опять из-за погрешностей округления значение λ_{min} тоже находится не точно. Поэтому можно сделать шаг λ_{step} не фиксированным, а динамическим. Пусть имеется некоторый шаг λ_{step} . Находим λ_{min} по формуле (5.9). Тогда, если $\frac{\lambda_{\text{min}}}{\lambda_{\text{step}}} \in (0, \varsigma] \cup [2 - \varsigma, +\infty)$, где ς — некоторое маленькое число, задаваемое пользователем, то полагаем

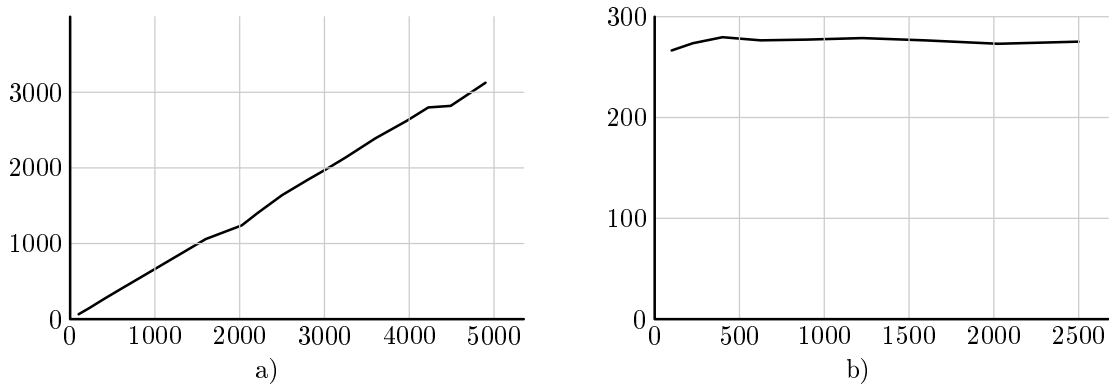


Рис. 4. а) Зависимость отношения времен вычисления $\text{grad } \widehat{M}^\alpha[\hat{z}]$ и $\widehat{M}^\alpha[\hat{z}]$ от N при $\tilde{N} = 100$.
 б) Зависимость отношения времен вычисления $\text{grad } \widehat{M}^\alpha[\hat{z}]$ и $\widehat{M}^\alpha[\hat{z}]$ от \tilde{N} при $N = 400$

$\lambda_{\text{step}} = \lambda_{\min}$ и снова вычисляем λ_{\min} . Следует, конечно, учитывать, что для точки $2\lambda_{\text{step}}$ соответствующая точка \hat{z} должна принадлежать множеству априорных ограничений, т.е. все ее координаты должны быть не отрицательны. Поэтому необходимо выбирать λ_{step} так, чтобы удовлетворялось последнее условие.

Отметим, что когда точка $\hat{z}^{(i)}$ минимизирующей последовательности близка к точке минимума функционала $\widehat{M}^\alpha[\hat{z}]$, то погрешности вычислений значений функционала могут сделать последующую минимизацию бесполезной, так как значения функционала в последующих точках $\hat{z}^{(p)}$, $p > i$, могут быть больше $\widehat{M}^\alpha[\hat{z}^{(i)}]$. Поэтому, наряду со стандартными условиями выхода из подпрограммы минимизации по числу итераций или по норме градиента, подпрограмма завершает свою работу, когда для одномерной минимизации функции $\sigma(\lambda)$ погрешности в вычислениях влияют на выпуклость функции.

Значение параметра регуляризации α определяет скорость сходимости минимизирующей последовательности. Функционал $\widehat{\Omega}[\hat{z}]$ является квадратичной функцией, а функционал $\widehat{I}[\hat{z}]$ не является. Поэтому ясно, что минимизирующая последовательность для $\widehat{\Omega}[\hat{z}]$ сходится быстрее, чем для $\widehat{I}[\hat{z}]$. На рис. 5 показана скорость сходимости для $\widehat{I}[\hat{z}]$ и $\widehat{\Omega}[\hat{z}]$ от числа итераций p (числа вычислений градиента).

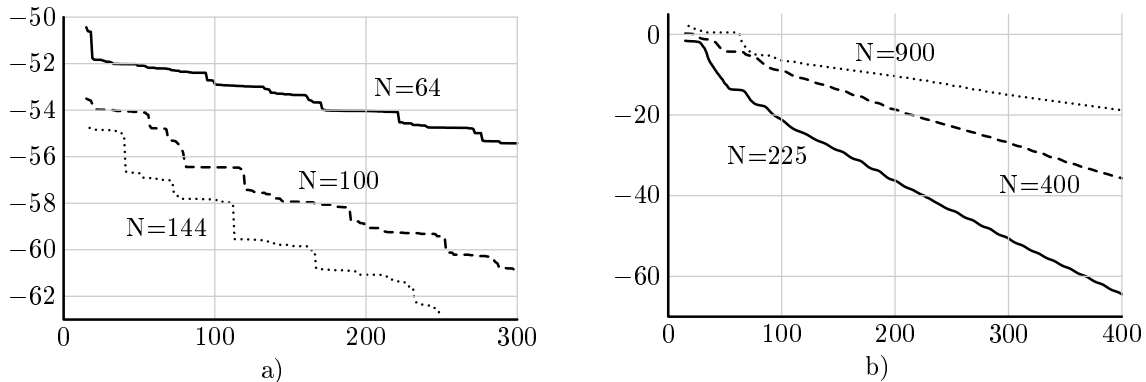


Рис. 5. а) Зависимость $\ln(\widehat{I}[\hat{z}])$ от числа шагов минимизации. б) Зависимость $\ln(\widehat{\Omega}[\hat{z}])$ от числа шагов минимизации

Для минимизации функционала $\widehat{I}[\hat{z}]$ использовались точная функция $\bar{z} = 1 + 0.3 \cos \varphi \sin \psi$ с параметрами **Задачи 1** и начальное приближение $z \equiv 1$. Для минимизации $\widehat{\Omega}[\hat{z}]$ — точная функция $\bar{z} \equiv 0$, начальное приближение $z = 1 + 0.3 \cos \varphi \sin \psi$. Из рис. 5 видно, что для рассматриваемых модельных задач скорости сходимости близки к экспоненциальным: Ae^{-ap} и Be^{-bp} ($a, b > 0$) для $\widehat{I}[\hat{z}]$ и $\widehat{\Omega}[\hat{z}]$ соответственно. Конечно, зависимость $\ln(\widehat{I}[\hat{z}])$ от числа итераций p , изображенная на рис. 5 а, не является линейной для последовательных шагов минимизации, но тенденция линейного убывания выполняется для большого числа шагов минимизации, т.е. прямая $-bp + \ln B$ является в некотором смысле асимптотой. Из рис. 5 также видно, что метод проекции сопряженных градиентов очень быстро переходит в окрестность точки минимума. Об этом свидетельствуют характерные колебания около прямых при малых p (на рисунке не отображены значения функций при $p < 20$, для которых наблюдается резкое убывание, зависящее от выбранного начального приближения).

Следует отметить также наблюдаемую зависимость констант a, b от размерности N : $a \sim N$ для $\widehat{I}[\hat{z}]$,

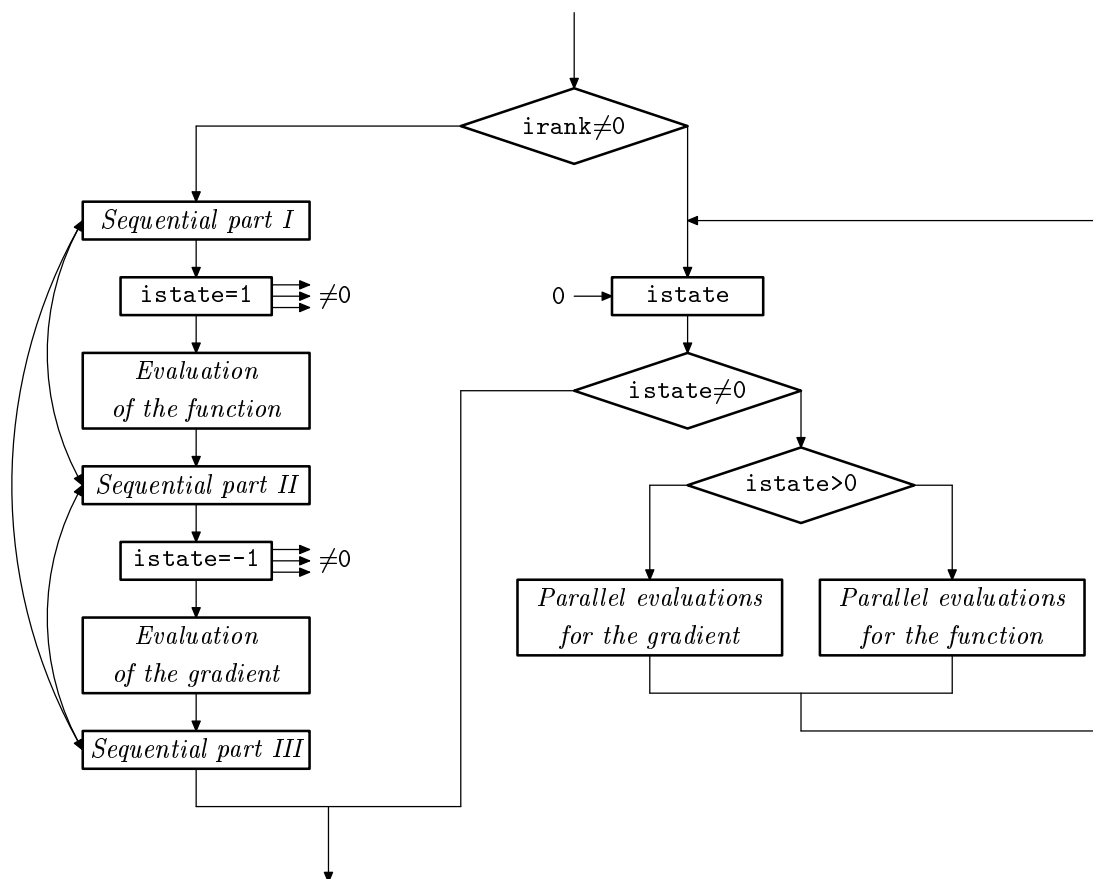


Рис. 6. Схема распараллеливания задачи минимизации для нескольких процессов

$b \sim 1/N$ для $\widehat{\Omega}[\hat{z}]$. Таким образом, при минимизации функционала $\widehat{M}^\alpha[\hat{z}]$ будет также наблюдаться экспоненциальное убывание значения функционала от числа итераций: Ce^{-cp} , $c > 0$. Зависимость константы c от размерности N будет определяться параметром регуляризации α : $c \sim 1/N$ при малых α и $c \sim N$ — при больших.

7. Распараллеливание задачи минимизации. При решении задачи минимизации методом проекции сопряженных градиентов на множество неотрицательных векторов необходимо вычислять значения функционала $\widehat{M}^\alpha[\hat{z}]$ и его градиента $\text{grad } \widehat{M}^\alpha[\hat{z}]$. Из формул (3.3), (3.4) для функционала $\widehat{M}^\alpha[\hat{z}]$ и (5.1), (5.2) для градиента $\frac{\partial}{\partial z_{ij}} \widehat{M}^\alpha$ следует, что для их нахождения можно применять многопроцессорные системы. Тогда задачу можно распараллелить, т.е. переписать программу таким образом, чтобы независимые части программы выполнялись на разных процессорах.

При решении модельных задач использовался вычислительный кластер Научно-исследовательского вычислительного центра МГУ. Кластер построен на базе двухпроцессорных узлов с процессорами Pentium III. Узлы объединены в двумерную решетку с помощью высокоскоростной сети SCI. В качестве служебной сети используется Fast Ethernet. Более подробную информацию о кластере можно найти в [4].

При написании программ для решения модельных задач картирования распределения химических элементов по поверхностям звезд на многопроцессорных компьютерах использовалась библиотека MPI (Message Passing Interface) [5]. В качестве языка программирования был выбран Fortran 90.

Исходная задача решается с использованием `nsize` (`nsize > 2`) параллельных процессов с номерами $0, \dots, \text{nsize} - 1$, выполняющихся на отдельном процессоре и взаимодействующих при необходимости друг с другом. Нулевой процесс (процесс с номером 0) выполняет все нераспараллеливаемые действия: чтение из файла, запись файла, формирование массивов. Распараллеливание происходит только при минимизации функционала $\widehat{M}^\alpha[\hat{z}]$: до и после минимизации ненулевые процессы не задействованы.

Все процессы содержатся в группе с предопределенным идентификатором `mpi_comm_world`. Каждый процесс имеет номер `irank`. На рис. 6 показана схема распараллеливания при решении задачи минимизации

ции. Для нулевого процесса имеются три взаимодействующих между собой блока, в которых происходит минимизация искомого функционала $\widehat{M}^\alpha[\hat{z}]$. Когда необходимо вычислить значения функционала $\widehat{M}^\alpha[\hat{z}]$, его градиента $\text{grad} \widehat{M}^\alpha[\hat{z}]$ или выйти из подпрограммы, то нулевой процесс передает управляющую переменную `istate` всем процессам с помощью подпрограммы `mpi_bcast` из библиотеки MPI:

```
call mpi_bcast(istate,1,mpi_integer,0,mpi_comm_world,ierr)
```

Если значение переменной `istate` равно нулю, то все процессы завершают свою работу и выходят из подпрограммы минимизации. Если же переменная `istate` положительна, то происходит вычисление функционала $\widehat{M}^\alpha[\hat{z}]$, а если отрицательна, то градиента $\text{grad} \widehat{M}^\alpha[\hat{z}]$. При этом каждый ненулевой процесс вычисляет значения, которые передает нулевому процессу для последующего формирования нулевым процессом значений функционала или градиента.

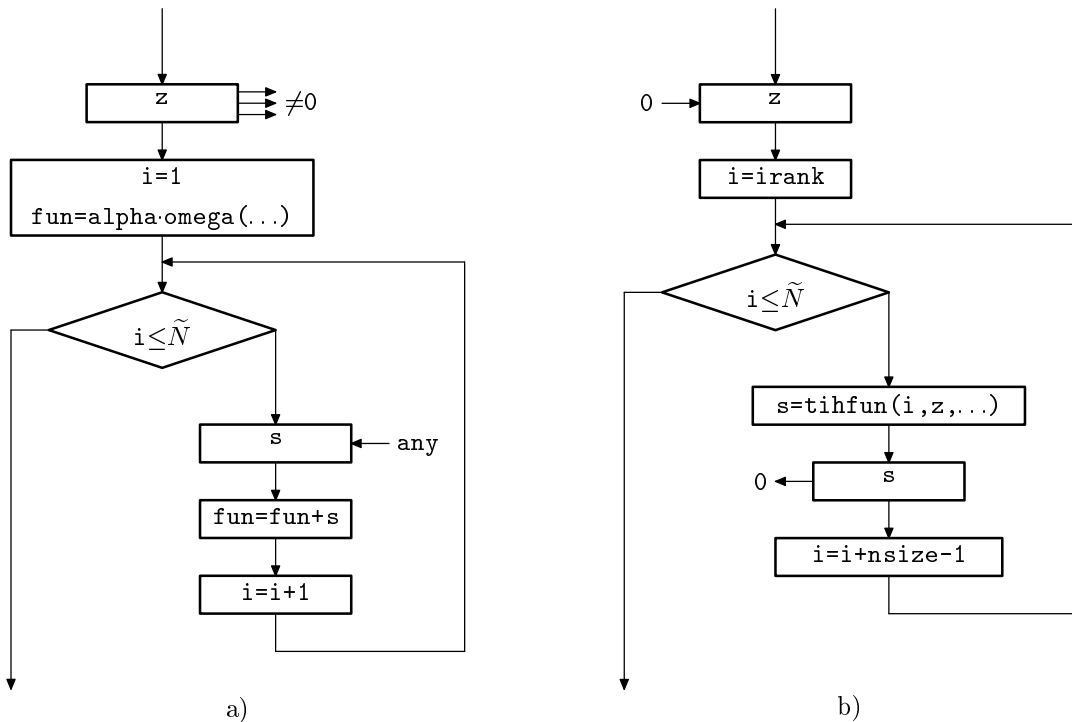


Рис. 7. а) Схема вычисления значения функционала для нулевого процесса. б) Схема работы ненулевого процесса при вычислении значения функционала

Рассмотрим подробнее задачу вычисления значения функционала $\widehat{M}^\alpha[\hat{z}]$ в некоторой точке \hat{z} . В связи с простым видом функционала $\widehat{\Omega}[\hat{z}]$ его вычисление производим с помощью одного (нулевого) процесса, используя функцию `omega(...)`. Вычисление функционала $\widehat{J}[\hat{z}]$ можно распараллелить, поставив для каждого процесса задачу вычисления квадрата выражения, стоящего под знаком двойной суммы в (3.3). Таким образом, нулевой процесс передает всем остальным процессам вектор \hat{z} (переменная `z` в схеме на рис. 7 а). После этого каждый ненулевой процесс вычисляет квадрат выражения для заданных переменных `k` и `l`. Все многомерные массивы можно записать как одномерные, поэтому в схеме переменные `k` и `l` из (3.3) заменяются переменной `i`, изменяющейся от 1 до `N`. Вычисление квадрата производится с помощью функции `tihfun(i,z,...)`, причем каждый процесс выполняет вычисления для своих значений `i`. Конечно, может оказаться, что к некоторому моменту времени некоторые процессы вычисляют все свои функции `tihfun(i,z,...)`, а другие — вычисляют только половину. Тогда необходимо применять другую схему распараллеливания, которая загружает процессы вычислениями по мере их освобождения. Для рассмотренных модельных задач времена вычислений значений функций `tihfun(i,z,...)` практически не зависели от `i`, т.е. предложенная схема оказалась достаточно быстрой. После вычисления значения функции `tihfun(i,z,...)` это значение с помощью функции `mpi_send` передается нулевому процессу, где происходит суммирование:

```
call mpi_send(s,1,mpi_double_precision,0,i,mpi_comm_world,ierr)
```

Нулевой процесс получает значения переменных `s` с помощью функции `mpi_recv`:

```
call mpi_recv(s,1,mpi_double_precision,mpi_any_source,mpi_any_tag,mpi_comm_world,istatus,
             ierr)
```

Видно, что нулевой процесс суммирует переменные s в том порядке, в котором они поступают. Именно в этом и заключается характерная особенность при вычислении значения функционала $\widehat{M}^\alpha[\hat{z}]$. Переменные s являются вещественными числами двойной точности примерно одного порядка. Однако при их суммировании происходит округление. Так как переменные s могут поступать в разном порядке от разных процессов, то из-за округления значения функционала $\widehat{M}^\alpha[\hat{z}]$ при повторных вычислениях отличаются в последних или предпоследних знаках. Но при увеличении числа итераций оказывается, что при числе шагов минимизации порядка 100 различие в значениях функционалов достигает нескольких процентов при повторных вычислениях. Более того, может оказаться, что для одного повторного вычисления найден минимум (выход по норме градиента), для второго повторного вычисления минимум еще ищется, а для третьего произошел выход из-за нарушения выпуклости минимизируемой функции.

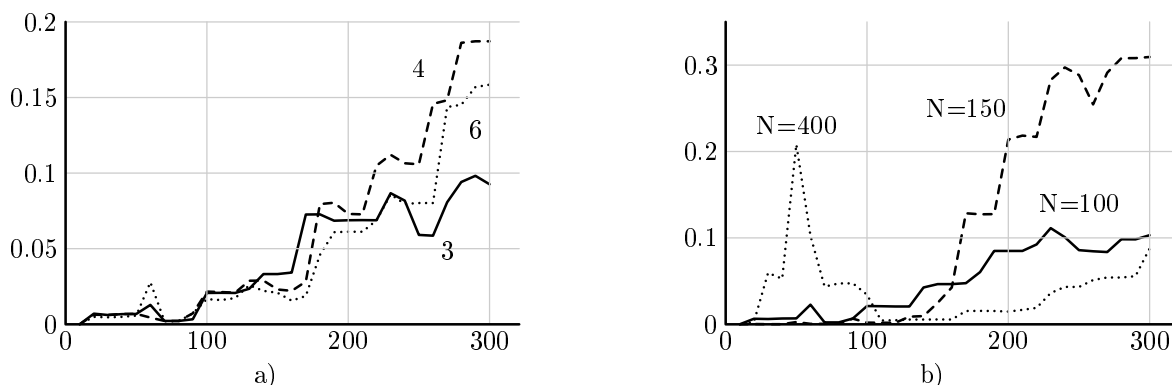


Рис. 8. Зависимость отношений ширины разброса значений функции к средним значениям при повторных вычислениях от числа итераций а) при различном числе процессов ($N = 100$) и б) при различных размерностях (8 процессов)

При решении модельных задач для каждой итерации были найдены минимальные и максимальные значения функционала $\widehat{M}^\alpha[\hat{z}]$ при повторных вычислениях. На рис. 8 показаны зависимости отношений ширины разброса полученных значений к средним значениям от числа итераций и размерности N . Производились 15 повторных вычислений. Для построенных зависимостей характерно в среднем возрастание при увеличении числа итераций. Но в силу того, что при каждом повторном вычислении минимизирующие последовательности сходятся к одной точке, то возможны также и резкие “провалы” зависимостей (особенно на рис. 8 б для $N = 400$). Следует также отметить, что поведение зависимостей не определяется ни размерностью задачи, ни числом используемых процессов.

Таким образом, при решении задачи минимизации на многопроцессорных компьютерах нарушается важное свойство обычной (последовательной) программы: сохранение результата при повторных вычислениях. Если необходимо обязательное выполнение данного свойства, то проблема решается очень просто: нулевой процесс не сразу суммирует полученные значения, а сначала записывает их в некоторый массив; после получения всех значений вызывается подпрограмма для суммирования элементов массива, например, с использованием предварительной сортировки элементов. В большинстве случаев сохранение результата при повторных вычислениях может и не понадобиться: выход из программы по градиенту или в результате нарушения выпуклости минимизируемой функции происходит только вблизи точки минимума. Поэтому при повторных вычислениях точка минимума все равно находится, меняется только число итераций (время решения задачи минимизации).

Схема распараллеливания задачи вычисления градиента $\widehat{M}^\alpha[\hat{z}]$ изображена на рис. 9. Алгоритм распараллеливания тот же, что и при вычислении функционала $\widehat{M}^\alpha[\hat{z}]$. Но есть и отличие: каждый ненулевой процесс вычисляет k -ый элемент вектора градиента $\widehat{M}^\alpha[\hat{z}]$ (массив `grad` в схеме на рис. 9 а) и передает нулевому процессу вычисленный элемент s вместе с номером k этого элемента в массиве. Таким образом, при формировании массива `grad` не важно, в какой последовательности нулевой процесс получает элементы s , так как числа k точно определяют их позиции в массиве `grad`. Ясно, что градиент $\widehat{M}^\alpha[\hat{z}]$, найденный по предложенной схеме, сохраняет свое значение при повторном вычислении.

На представленных в статье схемах нулевой процесс рассматривается только как процесс, переда-

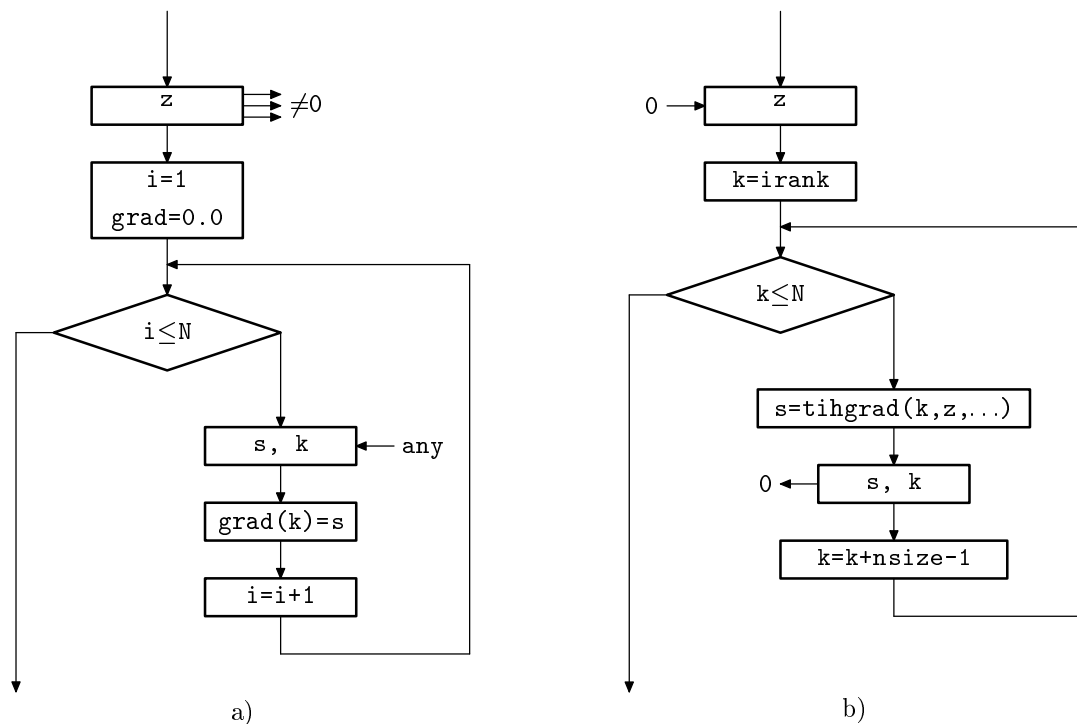


Рис. 9. а) Схема вычисления градиента функционала для нулевого процесса. б) Схема работы ненулевого процесса при вычислении градиента функционала

ющий и получающий информацию от других процессов. При вычислении значения функционала $\widehat{M}^\alpha[\hat{z}]$ и его градиента $\text{grad } \widehat{M}^\alpha[\hat{z}]$ можно задействовать и нулевой процесс в основных распараллеливаемых вычислениях. Для уменьшения накладных расходов, связанных приемом–передачей сообщений между процессами, а также для того, чтобы нулевой процесс работал полноценно, т.е. не был бы занят только обменом сообщениями, можно поступить следующим образом. Каждый процесс выполняет все вычисления, которые он должен сделать для нахождения значения функционала или его градиента, и только после этого передает данные нулевому процессу.

При решении задачи картирования распределения химических элементов по поверхностям звезд распараллеливание осуществлялось только при вычислении значений функционала Тихонова $\widehat{M}^\alpha[\hat{z}]$ и его градиента $\text{grad } \widehat{M}^\alpha[\hat{z}]$. Ясно, что все остальные вычисления производились последовательно. Поэтому при решении задачи на q процессорах время работы программы не уменьшалось точно в $q - 1$ раз по сравнению с нераспараллеленной программой (нулевой процесс согласно принятой схеме распараллеливания не участвует в основных параллельных вычислениях, он только передает и получает данные). Это и является одной из форм записи эмпирического закона Амдала, учитывающего наличие в программе нераспараллеливаемого кода. Но так как время работы последовательной части нашей программы при больших размерностях N и \tilde{N} пренебрежимо мало по сравнению с распараллеленным кодом, то можно считать, что время работы программы уменьшается в $q - 1$ раз на q процессорах.

Конечно, если даже рассмотреть распараллеленные блоки программы, например, вычисление функционала или его производной, то из-за того, что числа N и \tilde{N} в общем случае не являются кратными числу $q - 1$, окажется, что каждый процесс должен выполнить разный объем вычислений. Поэтому в некоторый момент времени все процессы уже закончат свою работу, а некоторые только приступят к последним вычислениям функций $\text{tihfun}(i, z, \dots)$, $\text{tihgrad}(k, z, \dots)$.

Для модельных задач вычислялось выражение $\tau_q = \frac{t_q}{t_2}(q - 1) - 1$, где q — число процессоров и t_q — время вычисления градиента для q процессоров. Как и следовало ожидать, при увеличении числа N числа τ_q стремятся к нулю: $\tau_8 = 3.26$ при $N = 100$, $\tau_8 = 0.71$ при $N = 900$, $\tau_8 = 0.46$ при $N = 3600$.

О временах, необходимых для решения задачи, можно судить по рис. 10, на котором показаны зависимости времен вычисления градиента $\text{grad } \widehat{M}^\alpha[\hat{z}]$ от размерностей N и \tilde{N} решаемой задачи для $q = 2$ (нераспараллеленная программа). Функция на рис. 10 хорошо аппроксимируется функцией $1.38 \cdot 10^{-4} N^2$,

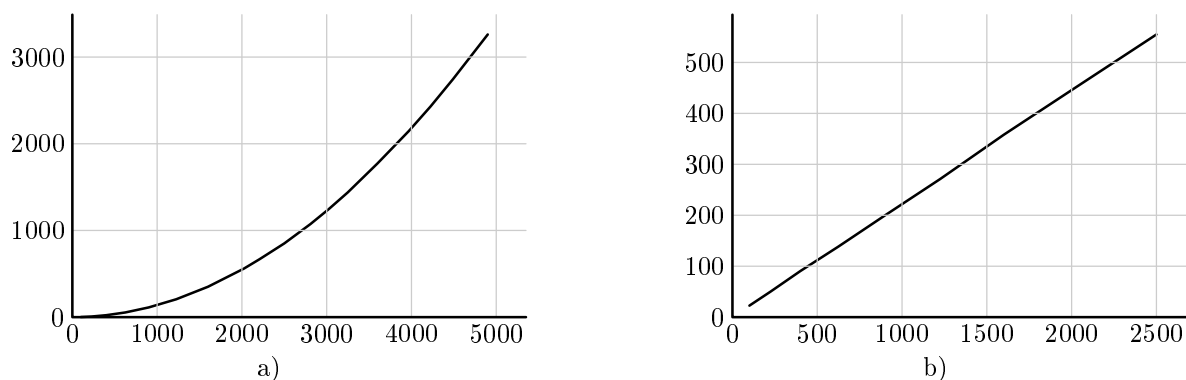


Рис. 10. а) Время вычисления (в секундах) градиента $\text{grad } \widehat{M}^\alpha[\hat{z}]$ от размерности N при $\tilde{N} = 100$. б) Время вычисления (в секундах) градиента $\text{grad } \widehat{M}^\alpha[\hat{z}]$ от размерности \tilde{N} при $N = 400$

а на рис. 10 б — функцией $0.221 \tilde{N}$. Таким образом, общая зависимость времени вычисления градиента от размерностей N и \tilde{N} имеет вид: $t_{\text{grad}} = 1.387 \cdot 10^{-6} N^2 \tilde{N}$ (в секундах). Для времени вычисления функции $\widehat{M}^\alpha[\hat{z}]$ получаем: $t = 2.128 \cdot 10^{-6} N \tilde{N}$ (в секундах).

Применение многопроцессорных компьютеров при решении задачи картирования распределения химических элементов на поверхности звезд позволяет существенно сократить время работы программы. Это очень важно, когда вместо нескольких суток работы обычной программы получаем всего несколько часов работы распараллеленной программы на вычислительном кластере.

Авторы благодарят междисциплинарный научный проект МГУ № 31 “Магнитные поля и турбулентность в космосе”.

СПИСОК ЛИТЕРАТУРЫ

1. Гончарский А.В., Черепашук А.М., Ягола А.Г. Некорректные задачи астрофизики. М.: Наука, 1985.
2. Тихонов А.Н., Леонов А.С., Ягола А.Г. Нелинейные некорректные задачи. М.: Наука, 1995.
3. Гончарский А.В., Степанов В.В., Хохлова В.Л., Ягола А.Г. Восстановление локальных профилей линий по наблюдаемому в спектре Ар-звезд // Письма в Астрон. журн. 1977. **3**, № 6. 278–282.
4. Вычислительный кластер НИВЦ МГУ (<http://parallel.ru/cluster>).
5. MPI: A Message-Passing Interface standard // The Message Passing Interface Forum, Version 1.1, June 12, 1995 (<http://www.mpi-forum.org>).

Поступила в редакцию
15.01.2002