УДК 004.272.2+536.77+537.622.4

Термодинамическая модель фазового равновесия многокомпонентных сплавов на основе Fe–Cr–Co и схема организации вычислений в ее рамках

© Н.А. Беляков, Б.Е. Винтайкин

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Предложен подход к организации вычислений в рамках термодинамической модели, описывающей фазовое равновесие в многокомпонентных магнитных сплавах на основе системы Fe–Cr–Co. Приведено краткое введение в предметную область решаемой задачи, описание основных составных частей модели, подробно описана схема вычислений.

Ключевые слова: *параллельные вычисления, термодинамическое моделирование, многокомпонентные сплавы, система Fe–Cr–Co, объектно-ориентированное проектирование, программное обеспечение, фазовое равновесие.*

Введение. В различных областях современных науки и техники постоянно возникают практически важные задачи, точное аналитическое решение которых значительно затруднено или невозможно, а числовое решение с необходимой точностью требует колоссального объема вычислений. Это может быть связано, например, с громоздкостью функциональных зависимостей, выражающих предметную область задачи, с большим числом измерений, параметров и т. д.

С подобными задачами приходится постоянно сталкиваться, например при разработке новых многокомпонентных материалов, обладающих сложной совокупностью заданных свойств. Большинство практически важных материалов представляют собой многокомпонентные многофазные системы со сложной структурой. На свойства таких систем оказывает влияние огромное число факторов, как внешних, так и внутренних. Все это приводит к невозможности исследований таких материалов с применением лишь натурных экспериментов и в разумные сроки. Поэтому в этой области все чаще используют численное моделирование систем. Одним из таких подходов, получившим широкое применение, является термодинамическое моделирование [1]. Этот подход, основанный на моделировании термодинамических характеристик системы и применении знаний о связи структуры со свойствами материала, позволяет численно моделировать процессы, протекающие в материале, и прогнозировать результаты протекания фазовых превращений, формирования типов структуры материала и связанных с ними физических свойств. Но численные модели, описывающие большое число свойств многокомпонентных материалов и связей между ними, как правило, получают-

ся громоздкими и содержат множество параметров. Все это приводит к чрезвычайно большим объемам вычислений.

Для эффективного проведения вычислений в рамках таких моделей требуется постоянное наращивание производительности вычислительных средств. Экстенсивный путь повышения производительности вычислительных машин, связанный с увеличением скорости работы отдельных их элементов, сегодня практически себя исчерпал, что связано в том числе и с физическими ограничениями, накладываемыми конечностью скорости распространения электромагнитного поля в веществе, проблемами отвода теплоты и увеличения энергопотребления. Эти проблемы решаются, в частности, за счет уменьшения размеров отдельных элементов микросхем. Однако существуют серьезные ограничения на дальнейшую миниатюризацию и рано или поздно будет достигнут физический предел размера отдельного элемента в микросхемах. Вследствие этого увеличение быстродействия в настоящее время осуществляется в значительной степени за счет увеличения количества параллельно работающих функциональных элементов. Поэтому весьма актуальной является проблема организации параллельных вычислений в рамках моделей, описывающих сложные многокомпонентные системы.

Термодинамическая модель сплавов на основе системы Fe–Cr–Co. В данной работе рассмотрены многокомпонентные высокопрочные магнитно-жесткие сплавы на основе системы Fe–Cr–Co, обладающие уникальным сочетанием высоких магнитных и механических характеристик (прочность, пластичность, коррозионная стойкость) [2]. Формирование высоких магнитных свойств сплавов на основе системы Fe–Cr–Co происходит в результате спинодального распада [3] пересыщенного твердого раствора на основе ферромагнитного α-железа с объемно-центрированной кубической (ОЦК) решеткой на две изоморфные твердорастворные фазы: обогащенную (α_1) и обедненную (α_2) железом и кобальтом. Продуктом такого распада является модулированная наноструктура, представляющая собой частицы одной фазы, изолированные прослойками другой, или взаимопроникающие области фаз. Наиболее ярко выраженным магнитным свойствам отвечает структура, состоящая из значительно вытянутых вдоль одного направления образований ферромагнитной α_1-фазы (10…40 нм), изолированных прослойками парамагнитной матрицы α_2 и расположенных квазипериодически. Причем решетки фаз должны быть когерентно сопряжены [4]. Тип получаемой структуры во многом зависит от условий формирования сплава (термической обработки), от его начального состава, наличия легирующих элементов, которые при введении в сплав в относительно малых количествах (до 0,05 атомных долей) могут существенно изменить его свойства.

Для получения определенных свойств материала необходимо прежде всего знать состав фаз в состоянии фазового равновесия, а также их структуру. Ранее была разработана термодинамическая модель [5] многокомпонентных двухфазных сплавов, испытывающих расслоение с образованием ферромагнитной фазы, в частности сплавов на основе системы Fe–Cr–Co. В рамках этой модели можно вычислять равновесные параметры сплавов с учетом структуры фаз, возможных фазовых превращений, влияния легирующих элементов и других факторов, оказывающих воздействие на свойства материалов. На основании этой модели можно выбирать параметры и условия термической обработки сплавов для получения требуемых свойств, предварительно подбирать составы исходных сплавов, количество легирующих элементов, строить фазовые диаграммы и т. д. Однако эффективная схема вычислений в рамках этой модели отсутствовала.

Моделируемой термодинамической величиной в модели [5] является термодинамический потенциал – изменение свободной энергии (энергии Гельмгольца) при образовании (*formation*) моля двухфазного N-компонентного сплава:

$$\begin{aligned}\Delta_f F_{2p,T}\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{v}\right) = \nu_1 \Delta_f F_T\left(\mathbf{x}^{(1)}\right) + \\ + \nu_2 \Delta_f F_T\left(\mathbf{x}^{(2)}\right) + \Delta_f F_{e2p}\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{v}\right),\end{aligned} \tag{1}$$

где $\mathbf{x}^{(p)} = \left(x_1^{(p)}, x_2^{(p)}, \ldots, x_i^{(p)}, \ldots, x_N^{(p)}\right)^{\mathrm{T}}$ – вектор концентраций компонентов фазы p; $\mathbf{v} = (\nu_1, \nu_2)^{\mathrm{T}}$ – вектор объемных долей фаз; ν_p – объемная доля фазы p; T – абсолютная температура (параметр задачи); $\Delta_f F_T(\mathbf{x}^{(p)})$ – свободная энергия образования фазы p, $p = 1, 2$,

$$\Delta_f F_T\left(\mathbf{x}^{(p)}\right) = \Delta_f F_{c,T}\left(\mathbf{x}^{(p)}\right) + \Delta_f F_{m,T}\left(\mathbf{x}^{(p)}\right). \tag{2}$$

Здесь

$$\begin{aligned}\Delta_f F_{c,T}\left(\mathbf{x}^{(p)}\right) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \frac{x_i^{(p)}\sum_{k=0}^{r} a_k T^k + x_j^{(p)}\sum_{k=0}^{s} b_k T^k}{x_i^{(p)} + x_j^{(p)}} x_i^{(p)} x_j^{(p)} + \\ + RT\sum_{i=1}^{N} x_i^{(p)} \ln\left(x_i^{(p)}\right) + \sum_{i=1}^{N} x_i^{(p)} \sum_{k=1}^{t} c_k T^k\end{aligned} \tag{3}$$

– так называемый химический вклад в свободную энергию образования фазы p, обусловленный тепловым эффектом и изменением энтропии при образовании одной фазы сплава. Логарифмические слагаемые в соотношении (3) выражают энтропию образования моля иде-

альных твердых растворов. Отклонение раствора от идеальности описывает дробное выражение, входящее в двойную сумму: оно включает аппроксимации так называемых энергий смешения, выражающих тепловой эффект и избыточное (по сравнению с идеальным) изменение энтропии системы. Крайняя правая сумма в выражении (3) описывает устойчивость различных кристаллографических модификаций элементов, входящих в твердый раствор, один относительно другого и также является аппроксимацией экспериментальных данных (для чистых компонентов) [5]. Коэффициенты полиномов a_k, b_k и c_k являются параметрами модели (главным образом эмпирическими). Их значения приведены в работе [5]. Степени полиномов r, s и t не превышают четырех, R = 8,314 Дж/моль – универсальная газовая постоянная.

Слагаемое $\Delta_f F_{m,T}(\mathbf{x}^{(p)})$ в выражении (2) представляет собой вклад энергии магнитного упорядочения в свободную энергию образования твердого раствора:

$$\Delta_f F_{m,T}\left(\mathbf{x}^{(p)}\right)=\begin{cases} F_1\left(\mathbf{x}^{(p)}\right)\left(\dfrac{T}{T_c\left(\mathbf{x}^{(p)}\right)}\right)^{n+1}-F_{0,T}\left(\mathbf{x}^{(p)}\right), & T\le T_c\left(\mathbf{x}^{(p)}\right); \\ F_1\left(\mathbf{x}^{(p)}\right)-A\left(\mathbf{x}^{(p)}\right)\left(T-T_c\left(\mathbf{x}^{(p)}\right)\right)+ & \\ +A_1\left(\mathbf{x}^{(p)}\right)m^2T_c\left(\mathbf{x}^{(p)}\right)\left(1-\exp\left(-\dfrac{T-T_c\left(\mathbf{x}^{(p)}\right)}{mT_c\left(\mathbf{x}^{(p)}\right)}\right)\right)- & \\ -F_{0,T}\left(\mathbf{x}^{(p)}\right), & T>T_c\left(\mathbf{x}^{(p)}\right), \end{cases} \tag{4}$$

и вычисляется с помощью зависимостей вклада магнитного упорядочения в теплоемкость системы в области температуры Кюри $T_c(\mathbf{x}^{(p)})$ сплава. Эти зависимости получены как в результате экспериментальных исследований, так и в рамках моделей магнитного упорядочения [5]. В выражении (4)

$$F_1\left(\mathbf{x}^{(p)}\right)=\frac{A_1\left(\mathbf{x}^{(p)}\right)T_c\left(\mathbf{x}^{(p)}\right)}{n\left(n+1\right)};$$

$$F_{0,T}\left(\mathbf{x}^{(p)}\right)=\frac{A_1\left(\mathbf{x}^{(p)}\right)T_c\left(\mathbf{x}^{(p)}\right)}{n+1}+A_1\left(\mathbf{x}^{(p)}\right)m\left(1+m\right)T_c\left(\mathbf{x}^{(p)}\right)-TA\left(\mathbf{x}^{(p)}\right);$$

$$A_1\left(\mathbf{x}^{(p)}\right)=\frac{A\left(\mathbf{x}^{(p)}\right)T_c\left(\mathbf{x}^{(p)}\right)}{1/n+m}.$$

Зависимость температуры Кюри фазы p от состава аппроксимируется выражением вида

$$T_c\left(\mathbf{x}^{(p)}\right) = T_{c(Fe)} + \sum_{i=1}^{N}\left(e_i x_i^{(p)} + f_i\left(x_i^{(p)}\right)^2\right).$$

Концентрационная зависимость изменения энтропии твердого раствора при магнитном фазовом переходе

$$A\left(\mathbf{x}^{(p)}\right) = A_{Fe}\sum_{i=1}^{N} h_i x_i^{(p)}.$$

В выражении (4) и в сопутствующих выражениях e_i, f_i, h_i, m, n ($i = 1, \ldots, N$) – параметры модели [5].

В выражении (1) третье слагаемое $\Delta_f F_{e2p}(\mathbf{x}^{(p)}, T)$ определяет вклад энергии упругих деформаций когерентно сопряженных фаз (т. е. кристаллические плоскости фаз переходят друг в друга непрерывным образом) в свободную энергию образования двухфазного сплава. Необходимость введения этой поправки возникает в случаях, когда разность параметров решеток фаз α_1 и α_2 достаточной велика, что достигается легированием сплава такими элементами, как молибден и вольфрам, которые при спинодальном распаде распределяются преимущественно в парамагнитную фазу. При этом в результате образуется отмеченная выше квазипериодическая модулированная структура, состоящая из вытянутых вдоль кристаллографического направления [001] («игольчатых») выделений фазы α_1, изолированных тонкими прослойками фазы α_2. Систематическое описание поправки $\Delta_f F_{e2p}(\mathbf{x}^{(p)}, T)$ можно найти в работе [4], оно достаточно громоздко и выходит за рамки данной работы.

На аргументы функции (1) наложены следующие естественные ограничения, которые прежде всего выражают закон сохранения количества вещества при фазовых переходах, а также условия нормировки величин:

$$x_i^{(p)} \in (0,\ 1); \quad \nu_p \in (0,\ 1); \quad \sum_{i=1}^{N} x_i^{(p)} = 1; \quad \nu_1 + \nu_2 = 1, \ \ p = 1, 2. \quad (5)$$

Кроме того, концентрации компонентов в фазах в состоянии термодинамического равновесия связаны друг с другом и с объемными долями соответствующих фаз известным в физической химии правилом рычага, фактически являющимся выражением закона сохранения массы. В случае двухфазной системы это правило имеет вид

$$\frac{\nu_2}{\nu_1} = \frac{x_i^{(0)} - x_i^{(1)}}{x_i^{(2)} - x_i^{(0)}}, \qquad (6)$$

где $\mathbf{x}^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, \ldots, x_i^{(0)}, \ldots, x_N^{(0)}\right)^{\mathrm{T}}$ – один из входных параметров задачи – вектор концентраций исходного сплава, который распадается на две фазы.

Свободная энергия образования твердого раствора (1) в состоянии, соответствующем фазовому равновесию системы, достигает минимума в пространстве параметров термодинамического равновесия – составов всех фаз и объемных долей фаз ($\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, $\boldsymbol{\nu}$) – с учетом ограничений (5) и (6) при фиксированной температуре T (рассматриваемой как параметр задачи). Причем равновесный фазовый состав сплава существенно зависит от состава исходного твердого раствора $\mathbf{x}^{(0)}$, распадающегося на две фазы [6]. Поэтому вектор концентраций компонентов в исходном твердом растворе также рассматривается как параметр задачи и должен быть зафиксирован на каждом этапе ее решения. Таким образом, формально вычисление равновесных составов и объемных долей фаз в рамках модели предполагает решение задачи локальной минимизации функции (1) в пространстве составов всех фаз и объемных долей фаз с учетом ограничений (5) и (6):

$$\Delta_f F_{2p,T}\left(\tilde{\mathbf{x}}^{(1)}, \tilde{\mathbf{x}}^{(2)}, \tilde{\boldsymbol{\nu}}\right) = \min_{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \boldsymbol{\nu}} \Delta_f F_{2p,T}\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \boldsymbol{\nu}\right)\Big|_{\substack{T=\mathrm{const} \\ \mathbf{x}^{(0)}=\mathrm{const}}}, \qquad (7)$$

где $\tilde{\mathbf{x}}^{(1)}$, $\tilde{\mathbf{x}}^{(2)}$ – соответственно искомые составы фаз α_1 и α_2 в состоянии равновесия; $\tilde{\boldsymbol{\nu}} = (\tilde{\nu}_1, \tilde{\nu}_2)^{\mathrm{T}}$ – искомый вектор объемных долей этих фаз.

Если в целевую функцию (1) включить ограничения (5) и (6), которые, в частности, связывают концентрации элементов и объемные доли обеих фаз, а также концентрации элементов каждой из фаз друг с другом, то задачу (7) можно свести к виду [7]

$$\Delta_f F_{2p,T}\left(\tilde{\mathbf{x}}^{(1)}, \tilde{\nu}_1\right) = \min_{\mathbf{x}^{(1)}, \nu_1} \Delta_f F_{2p,T}\left(\mathbf{x}^{(1)}, \nu_1\right)\Big|_{\substack{T=\mathrm{const} \\ \mathbf{x}^{(0)}=\mathrm{const}}}, \qquad (8)$$

причем здесь $\mathbf{x}^{(1)} = \left(x_1^{(1)}, x_2^{(1)}, \ldots, x_i^{(1)}, \ldots, x_{N-1}^{(1)}\right)^{\mathrm{T}}$ – вектор $N-1$ независимых концентраций фазы α_1 (одно из ограничений (5) позволяет выразить концентрацию одного из компонентов фазы через концентрации остальных). Таким образом, всего имеется N параметров минимизации: $N-1$ независимая концентрация компонентов фазы α_1 и объемная доля этой фазы. Далее для краткости обозначим набор параметров минимизации через $\mathbf{x}$ (т. е. $\mathbf{x} = (\mathbf{x}^{(1)}, \nu_1)$), а целевую функцию – через $F = F(\mathbf{x})$.

Можно показать [7], что вследствие симметрии области минимизации, выраженной ограничениями (5), (6), а также симметрии самой целевой функции (1), для нахождения параметров устойчивого равновесия системы достаточно найти локальный минимум целевой функции, ближайший к точке исходного состава $\mathbf{x}_0$ [7].

Ввиду большого числа параметров, в том числе эмпирических, исчерпывающее аналитическое исследование рассматриваемой целевой функции весьма затруднено, поэтому затруднено и аналитическое решение задачи минимизации. Для решения задачи (8) использовали численные методы. Выбор подходящего метода минимизации в данном случае представляет собой весьма трудоемкую задачу оптимизации, где в качестве параметра оптимизации можно использовать, например, число вычислений значения целевой функции, необходимое для нахождения ее минимума с заданной точностью. Распространенным способом оценки эффективности методов минимизации и корректности получаемого результата в случаях, когда аналитические оценки затруднены, являются вычислительные эксперименты и сравнительный анализ алгоритмов, реализующих методы минимизации, по результатам таких экспериментов. Подобное исследование проведено в работе [7]. Выбран модифицированный метод градиентного спуска с дроблением шага [8]. В процессе нахождения минимума строится так называемая релаксационная последовательность $\{\mathbf{x}_k\}$, осуществляющая переход от точки $\mathbf{x}_{k-1}$ к точке $\mathbf{x}_k$ области допустимых значений параметров минимизации таким образом, что для целевой функции $F(\mathbf{x})$ выполняется условие монотонности:

$$F_k = F\left(\mathbf{x}_k\right) \le F\left(\mathbf{x}_{k-1}\right) = F_{k-1}, \qquad k = 0,\ 1,\ 2,\ \ldots \tag{9}$$

при этом движение от точки $\mathbf{x}_{k-1}$ к точке $\mathbf{x}_k$ осуществляется в направлении, противоположном градиенту целевой функции в точке $\mathbf{x}_{k-1}$, т. е. в направлении наибольшего убывания функции при бесконечно малом движении из этой точки. Выбранный подход для вычисления k-го приближения использует не только $(k-1)$-е, но и $(k-2)$-е приближение, что позволяет лучше приспосабливаться к сложному рельефу целевой функции:

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \beta_k \nabla F\left(\mathbf{x}_{k-1}\right) + \gamma_k \left(\mathbf{x}_{k-1} - \mathbf{x}_{k-2}\right).$$

Здесь β_k – переменный шаг минимизации; γ_k – так называемый коэффициент вязкости ($0 < \gamma_k < 1$), помогающий ускорить сходимость алгоритма в случае «овражного» или «котловинного» рельефа [8]. Для каждого приближения минимизации k на основании предыдущих

приближений выбирается некоторое начальное значение шага $\beta_k = \beta_0$, далее осуществляется движение в направлении, противоположном градиенту целевой функции. Если условие монотонности (9) не нарушено, то осуществляется переход к следующему приближению, а если нарушено, то производится дробление шага с некоторым заданным коэффициентом с последующей попыткой осуществить спуск.

Для учета ограничений (5) и (6) использована разновидность метода штрафных функций [8]. Целевая функция представлена в виде

$$\hat{F}(\mathbf{x}) = \begin{cases} F(\mathbf{x}), & \mathbf{x} \in \Omega; \\ +\infty, & \mathbf{x} \notin \Omega, \end{cases}$$

где Ω – область допустимых значений, задаваемая ограничениями (5) и (6). В реальной вычислительной системе вместо $+\infty$, естественно, использовалось максимальное значение, которое может принимать целевая функция (вследствие аппаратных ограничений вычислительной системы).

Схема параллельных вычислений. Исходная задача может быть сформулирована в виде функции $\Phi_{\mathbf{p}}(\mathbf{x})$, где $\mathbf{x}$ – вектор аргументов, который в данном случае представляет собой и входной, и выходной параметры алгоритма, выражаемого этой функцией (на входе – начальное приближение алгоритма минимизации $\mathbf{x}_0$, на выходе – искомая точка минимума $\tilde{\mathbf{x}} = (\tilde{\mathbf{x}}^{(1)}, \tilde{\nu}_1)$), а $\mathbf{p}$ – вектор параметров задачи (включает состав исходного сплава $\mathbf{x}^{(0)}$ и температуру T: $\mathbf{p} = (\mathbf{x}^{(0)}, T)$). Элементы вектора $\mathbf{x}$ варьируются в процессе решения задачи, а вектор $\mathbf{p}$ остается постоянным:

$$\Phi_{\mathbf{p}}(\mathbf{x}) = \min_{\mathbf{x}} F_{\mathbf{p}}(\mathbf{x})\Big|_{\mathbf{p}=\text{const}}. \qquad (10)$$

Причем требуется многократно решать эту задачу для разных значений параметров $\mathbf{p}$. Решение задачи для каждого отдельного значения $\mathbf{p}$ представляет (в данном случае) последовательный процесс градиентной минимизации, требующий большого объема вычислений, вследствие громоздкости вида целевой функции. Но решение задачи для одного значения $\mathbf{p}$ не зависит от решения для другого значения, т. е. решать задачу для разных значений $\mathbf{p}$ можно параллельно в рамках SPMD-модели параллельных вычислений (Single Program/Multiple Data), где разные программные потоки выполняют разные части одной и той же программы с разными данными [9].

Для обеспечения максимальной независимости программной системы параллельной обработки от конкретной прикладной задачи (а следовательно, и возможности ее повторного использования для

решения других задач) при ее проектировании выбран объектно-ориентированный подход. В системе выделены три основных независимых модуля:

- модуль M_1, реализующий модель прикладной задачи (в данном случае это численная модель сплава);
- модуль M_2, осуществляющий решение прикладной задачи (в данном случае это локальная минимизация некоторой целевой функции);
- модуль M_3, организующий параллельные вычисления для решения прикладной задачи.

Связь между этими модулями осуществляется с помощью общих интерфейсов, что обеспечивает слабое зацепление между ними и возможность их использования независимо друг от друга, так как при таком подходе реализация каждого модуля не зависит от реализаций других.

Как видно из представленной декомпозиции задачи, уровень абстракции возрастает от модуля M_1 к модулю M_3: модуль M_1 представляет собой формулировку конкретной прикладной задачи, отражающую ее конкретную предметную область и сам по себе ничего не решает (это некоторая численная модель); область применения модуля M_2 может не ограничиваться моделью, представленной в M_1, и может включать в себя достаточно широкий класс задач локальной оптимизации; но задачи оптимизации, решаемые модулем M_2, – это лишь частный случай задач, решение которых может осуществлять модуль M_3. Кроме того, модуль M_1 инкапсулирует физическую составляющую данной работы, в модуле M_2 в основном сосредоточена вся математика и численные методы, а модуль M_3 заключает в себя программирование параллельной обработки данных.

Выбранная SPMD-модель организации параллельных вычислений хорошо соответствует архитектуре вычислительных систем с общей памятью. Подобные вычисления удобно проводить, в частности, и на широко распространенных в настоящее время настольных вычислительных машинах на базе многоядерного центрального процессора с 80 x 86-совместимой архитектурой. В таких системах в качестве сущности операционной системы, в рамках которой осуществляются вычисления, в данном случае лучше использовать потоки выполнения уровня ядра операционной системы (ОС), работающие в рамках единого процесса [9]. Каждый поток может практически полностью независимо от других производить вычисление функции (10) для заданного набора параметров **p** задачи.

Особенностью рассматриваемой прикладной задачи является то, что при разных значениях параметров **p** задачи расчеты в рамках второго модуля могут занимать достаточно различающиеся интервалы времени. Причем априорно невозможно установить, сколько времени будет выполняться какая-либо из частей задачи. Впрочем, такая

особенность может быть присуща и любой другой задаче. Поэтому простая статическая разбивка всего диапазона параметров на одинаковые по объему интервалы, число которых равно числу вычислительных ядер в системе, оказывается неэффективной, и требуется некоторая динамическая балансировка загрузки ядер. Такую балансировку (а также предварительную разбивку диапазона параметров, распределение работы между вычислительными потоками и сбор результатов) осуществляет основной компонент модуля M_3 – объект–*планировщик* потоков. Причем любая подобная организация работы потоков требует их синхронизации, но, как показывает практика, все издержки окупаются за счет более полной загрузки вычислительных ядер системы.

Один из способов такой балансировки – поделить всю задачу на достаточно мелкие части – *порции* (в пределе – на атомарные порции) и выдавать эти порции потокам по мере необходимости. На начальном этапе отдать каждому потоку по порции задачи и при завершении каждого потока выдавать ему следующую порцию из общего «набора порций» задачи, пока порции не закончатся (поток должен как-то сообщить планировщику о своем завершении). Под такой порцией в данном случае подразумевается некоторое число значений вектора параметров **p** задачи из заданного диапазона (очевидно, что атомарная порция – одно значение). Но возникает вопрос: насколько большими должны быть эти порции? Если они слишком большие, то это приведет к неравномерности загрузки потоков. Слишком маленькие порции (в пределе – атомарные) приведут к равномерной загрузке, но потребуют дополнительных затрат, прежде всего на организацию последовательного доступа к общему «набору порций» и другие синхронизации: в схеме появится слишком много вынужденных последовательных участков, что в соответствии с законом Амдала [9] приведет к снижению эффективности параллельной программы. Кроме того, при слишком маленьких порциях появляется и другая «статья дополнительных расходов»: возможная необходимость частого создания потоков выполнения (как сущностей ОС). Хотя потоки для своего создания требуют значительно меньше ресурсов, чем процессы, тем не менее слишком частое их создание может привести к снижению производительности (особенно это касается потоков уровня ядра, с которыми взаимодействует системный планировщик ОС в вытесняющем режиме – именно такие потоки рассматриваются в данной работе). Хотелось бы иметь схему планирования с динамической балансировкой загрузки, сочетающую достоинства статического разбиения задачи (редкое создание потоков ОС, минимум синхронизаций) и динамической схемы с очень малыми порциями (равномерная высокая загрузка потоков) и сводящую к некоторому минимуму их недостатки.

Для перебора значений параметров **p** и формализации понятий «задача» и «часть задачи» введена дополнительная абстракция – итератор, реализующая одноименный поведенческий шаблон объектно-ориентированного проектирования [10]. В частности, с помощью итераторов вычислительные потоки осуществляют перебор значений параметров задачи. Поскольку параметры задачи существенно зависят от ее предметной области, то для обеспечения независимости планировщика от конкретной прикладной задачи каждый итератор должен реализовывать некий общий интерфейс. Этот интерфейс должен предоставлять операцию доступа к следующей атомарной части задачи (в данном случае это отдельное значение вектора параметров **p**), назовем ее *next*, а также операцию проверки наличия оставшихся нерешенных частей задачи – *hasNext*. Планировщику потоков через ссылку на этот общий интерфейс передается итератор, включающий в себя весь диапазон параметров, что позволяет отвлечься от конкретной реализации итератора. Планировщик должен как-то разбить итератор на части, т. е. создать несколько итераторов, каждый из которых включает в себя некоторый поддиапазон параметров из общего диапазона, заключенного в передаваемый планировщику итератор. Для облегчения этой задачи планировщику включим в обобщенный интерфейс итератора операцию *split*, реализующую функциональность быстрого разбиения диапазона на части в заданном соотношении.

Предлагается следующая схема динамического планирования.

1. На этапе инициализации вычислений, перед их запуском, планировщик разбивает всю задачу на n равных порций с помощью последовательного вызова операции *split* над переданным ему итератором с соотношением $1/n$, где n – число вычислительных потоков, доступных планировщику (задается явно). Полученные итераторы передаются всем вычислительным потокам во владение.

2. Потоки запускаются на счет.

3. Когда поток завершается, объект потока извещает планировщика об этом (с помощью поведенческого паттерна Наблюдатель [10]).

4. Планировщик последовательно просматривает все еще работающие потоки в поисках потока, у которого на момент просмотра остается больше всего работы. Для этого в интерфейс итератора включена также операция *remaining*, возвращающая число атомарных частей задачи, оставшихся в итераторе. Перед тем как получить от каждого потока данные об оставшейся работе, планировщик устанавливает специальный флаг блокировки итератора в объекте вычислительного потока. Сам вычислительный поток при этом специально не приостанавливается и может продолжать вычисления. Приостановка потока произойдет, если он попытается обратиться к своему

итератору и взять очередную атомарную часть задачи. Если же флаг будет снят до того, как закончатся очередные вычисления функции (10), то вычислительный поток не заметит, что его опрашивал планировщик. Если у очередного просматриваемого потока работы осталось меньше, чем у найденного ранее «максимально загруженного» потока, то флаг блокировки в этом очередном потоке снимается и планировщик просматривает следующий поток. Если у очередного просматриваемого потока работы осталось больше, то с потока, который считался максимально загруженным до этого, снимается флаг блокировки, и этот поток извещается, что он может продолжать работу. Очередной просматриваемый поток теперь считается максимально загруженным – планировщик, не снимая с него флага блокировки, переходит к следующему потоку из набора и т. д., пока не будут просмотрены все потоки в наборе потоков.

5. Работа максимально загруженного потока делится планировщиком поровну с помощью операции *split* итератора, и половина отдается завершившемуся ранее потоку, инициировавшему весь этот процесс планирования, а половина остается у найденного потока.

6. Планировщик запускает на счет завершившийся ранее поток и снимает флаг блокирования с найденного максимально загруженного потока, извещая его при необходимости, что он может продолжать работу.

7. «Максимально загруженный» поток может быть и не найден во всем множестве потоков, если у каждого выполняющегося потока осталось атомарных частей задачи меньше некоторого заданного минимума (например, меньше двух). В этом случае планировщик ничего не делает, он ожидает завершения всех потоков.

На рис. 1 приведена упрощенная блок-схема алгоритма планирования (процесс планирования последовательный и выполняется в одном потоке – потоке планировщика, поэтому его алгоритм можно изобразить на блок-схеме), где i_max – индекс максимально загруженного потока в наборе потоков; r_max – оставшаяся работа (число оставшихся значений параметров у максимально загруженного потока); r_max0 – некоторая минимальная порция параметров, при которой деления не происходит (в пределе – атомарная порция); i_finished – индекс завершившегося потока, инициировавшего процедуру планирования (планировщик легко может его узнать); iter_lock(i) – функция блокировки итератора потока i; iter_unlock(i) – функция разблокировки итератора потока i; remaining(i) – функция определения оставшейся работы потока i; ir – оставшаяся работа у потока i; threadsCount – число потоков в наборе; iter_split(i) – функция разбиения итератора потока i пополам, возвращающая splitted_iter – итератор, содержащий в себе половину значений итератора потока i (число значений итератора в потоке i уменьшается соответственно); iter_set(i, iter) – функция установки итератора iter в поток i.

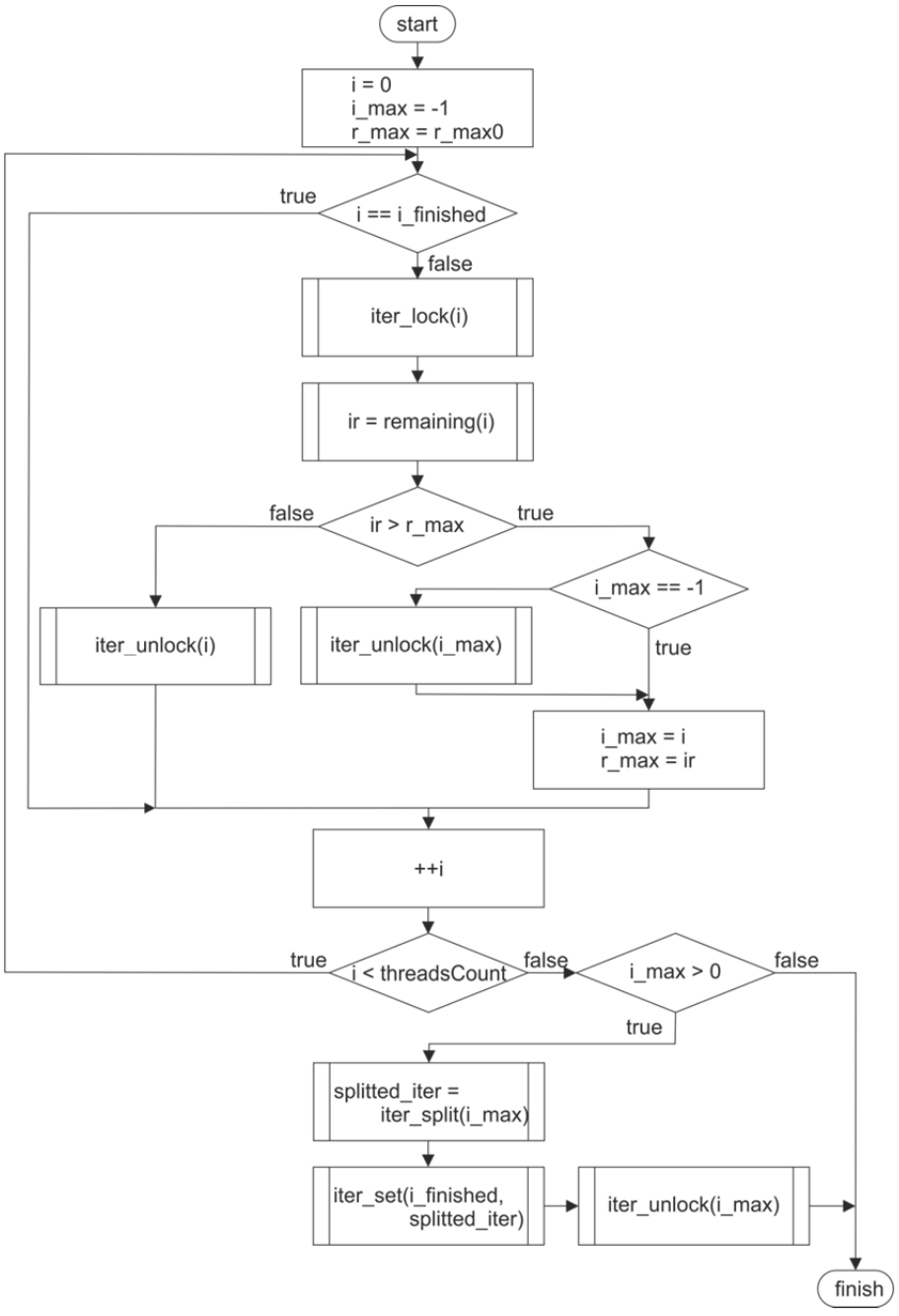

Рис. 1. Блок-схема алгоритма планирования

Для повышения производительности и минимизации использования различных средств синхронизации каждый вычислительный поток накапливает результаты вычислений в своем внутреннем контейнере. Сборкой результатов и помещением их в единый общий контейнер занимается планировщик по завершении работы всех потоков (по окончании вычислений или по их прерывании). Важно учи-

тывать, что использование приведенной схемы планирования приводит к заранее неопределенному порядку следования результатов. Например, при решении рассматриваемой прикладной задачи найденные точки фазового равновесия будут помещены в результирующий контейнер в некотором заранее неопределенном порядке. Впоследствии их можно будет как-нибудь упорядочить (по значениям некоторых параметров, например температуры).

Результаты. На рис. 2 приведены зависимости времени вычислений и ускорения от числа используемых потоков для решения задачи поиска параметров термодинамического равновесия в пятикомпонентных сплавах ($N = 5$) Fe–Cr–Co–Al–Nb, рассмотренных в работе [5],

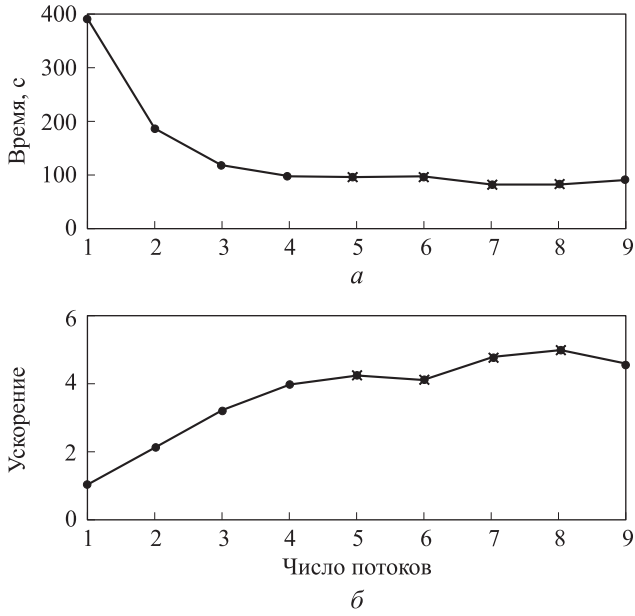

Рис. 2. Время вычислений (*а*) и ускорение (*б*) для рассматриваемого примера: сплавы Fe–0,16Cr–0,15Co–0,01Al–0,01Nb с точностью 10^{-5} в системе Intel Core i7 870 с Ubuntu Linux 12.04 x 64

со следующим набором входных параметров **p** задачи: температура T в диапазоне значений 773…973 K с шагом перебора 0,1 K, фиксированный состав исходного твердого раствора $\mathbf{x}^{(0)}$: $x^{(0)}_{(\mathrm{Co})} = 0{,}15$, $x^{(0)}_{(\mathrm{Cr})} = 0{,}16$, $x^{(0)}_{(\mathrm{Al})} = 0{,}01, x^{(0)}_{\mathrm{Nb}} = 0{,}01$ (атомные доли), т. е. всего имеется 201 атомарная часть задачи. В таблице приведены соответствующие числовые значения. В данном примере итератор параметров перебирает тот элемент вектора **p**, где хранится текущая температура, в заданном диапазоне и с заданным шагом, оставляя неизменными остальные его элементы. Также реализованы итераторы, перебирающие другие элементы вектора **p**, отвечающие за состав исходного сплава.

Таблица

Время вычислений и ускорение (для примера на рис. 2)

Число потоков	Среднее время вычислений *t*, с	Ускорение, отн. ед.
1	430,955	1,000
2	223,814	1,925
3	148,523	2,902
4	108,270	3,980
5	107,750	4,000
6	101,881	4,223
7	95,700	4,503
8	91,635	4,703
9	105,966	4,067

В качестве вычислительной системы использовалась рабочая станция на базе 80 x 86-совместимого центрального процессора Intel Core i7 870, имеющего четыре вычислительных ядра, каждое из которых поддерживает два потока (посредством технологии Intel HyperThreading), всего восемь логических ядер. Эксперименты проводились в операционной системе Ubuntu Linux 12.04 x 64. На рис. 2 отмечены усредненные по трем экспериментам значения. Потокам назначались маски соответствия (*affinity mask*) вычислительным ядрам таким образом, чтобы сначала использовались четыре физических ядра и только затем загружались логические ядра, предоставляемые технологией HyperThreading.

На рис. 2 видно, что вначале (при загрузке четырех физических ядер) ускорение растет практически линейно с угловым коэффициентом, близким к 1. Далее (при использовании технологии Hyper Threading) скорость роста существенно снижается и зависимость перестает быть линейной. Ускорение на четырех физических ядрах составило примерно 98,5 % от линейного. Также видно, что схема достаточно эффективно использует технологию Intel Hyper-Threading. Так, дополнительное ускорение на восьми потоках с задействованием технологии Hyper-Threading составляет примерно 19 % от линейного на четырех потоках.

На рис. 3 приведена вычисленная в рамках модели диаграмма, выражающая зависимость концентрации хрома в обеих фазах рассматриваемого двухфазного трехкомпонентного сплава от температуры (сечение фазовой диаграммы). Это один из результатов решения прикладной задачи в рассматриваемом частном случае. Исходный однородный сплав (α) становится неоднородным: он распадается при температурах ниже 973 K на две фазы: α_1 и α_2, которые отличаются концентрациями компонентов. Из диаграммы на рис. 3 следует, что концентрация хрома в фазе α_1 мала (а значит, она обогащена же-

лезом и кобальтом). Эта фаза обладает ферромагнитными свойствами. Фаза α_2, наоборот, содержит большое количество хрома и парамагнитна. Концентрация компонентов в фазе α_1 для каждого значения температуры находится из решения задачи (8), а концентрация компонентов в фазе α_2 – из соотношения (6).

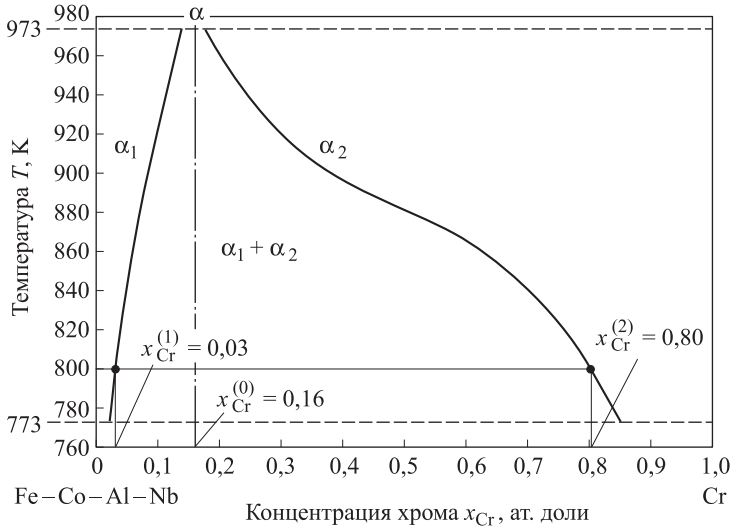

Рис. 3. Вычисленное вертикальное сечение фазовой диаграммы для системы Fe–0,16Cr–0,15Co–0,01Al–0,01Nb

Выводы. В работе описана термодинамическая модель фазового равновесия многокомпонентных магнитных сплавов на основе системы Fe–Cr–Co и предложена схема организации параллельных вычислений в ее рамках. На практически важных наборах входных данных для исследуемой прикладной задачи эта система обеспечивает ускорение вычислений на разных физических вычислительных ядрах, близкое к линейному. Так, на рабочей станции на базе четырехъядерного процессора Intel Core i7 870 ускорение на четырех потоках составляет около 98,5 % от линейного. Прибавка к ускорению за счет применения технологии Intel Hyper-Threading на восьми потоках четырехъядерного процессора составляет примерно 19 %.

ЛИТЕРАТУРА

[1] Kattner U.R. Thermodynamic Modeling of Multicomponent Phase Equilibria. *JOM: Journal of the Minerals, Metals and Materials Society*, 1997, vol. 49, № 12, pp. 14–19.

[2] Винтайкин Б.Е. Закономерности формирования структуры и магнитных свойств магнитно-жестких сплавов на основе Fe–Cr–Co. *Металловедение и термическая обработка металлов*, 1997, № 12, с. 12–14.

[3] Скрипов А.В., Скрипов В.П. Спинодальный распад. *Успехи физических наук*, 1979, т. 128, № 2, с. 193.

[4] Беляков Н.А., Винтайкин Б.Е. Исследование влияния энергии упругих деформаций когерентно-сопряженных фаз на фазовое равновесие в сплавах на основе системы Fe–Cr–Co методами термодинамического моделирования. *Вестник МГТУ. Естественные науки*, 2012, № 5, с. 65–74.

[5] Беляков Н.А., Винтайкин Б.Е. Термодинамическое моделирование формирования фазового равновесия в многокомпонентных сплавах, испытывающих расслоение с образованием магнитно-упорядоченной фазы. *Металловедение и термическая обработка металлов*, 2011, № 1 (667), с. 41–46.

[6] Гинье А., пер. с фр. под ред. Белова Н.В. *Рентгенография кристаллов*. Москва, Изд-во физ.-мат. лит., 1961, 604 с.

[7] Беляков Н.А. *Объектно-ориентированная система организации параллельных вычислений с приложением к задачам термодинамического моделирования*. LAP LAMBERT Academic Publishing, 2012, 160 с.

[8] Ахмеров Р.Р. *Методы оптимизации гладких функций*. [Электрон. ресурс]: электронный учебник: http://www.ict.nsc.ru/ru/textbooks/akhmerov/mo_unicode/index.html (дата обращения 10.04.2012).

[9] Воеводин В.В., Воеводин Вл.В. *Параллельные вычисления*. Санкт-Петербург, БХВ-Петербург, 2004, 608 с.

[10] Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. *Приемы объектно-ориентированного проектирования. Паттерны проектирования*. Санкт-Петербург, Питер, 2010, 366 с.

Статья поступила в редакцию 05.06.2013

Ссылку на эту статью просим оформлять следующим образом:

Беляков Н.А., Винтайкин Б.Е. Термодинамическая модель фазового равновесия многокомпонентных сплавов на основе Fe–Cr–Co и схема организации вычислений в ее рамках. *Инженерный журнал: наука и инновации*, 2013, вып. 8. URL: http://engjournal.ru/catalog/fundamentals/physics/1108.html

Беляков Никита Андреевич родился в 1984 г., окончил МГТУ им. Н.Э. Баумана в 2007 г. Магистр техники и технологии, аспирант кафедры «Физика» МГТУ им. Н.Э. Баумана. Автор 10 опубликованных работ. Области научных интересов: физика твердого тела, термодинамическое моделирование, параллельные вычисления. e-mail: nickbelyakov@mail.ru

Винтайкин Борис Евгеньевич родился в 1961 г., окончил МГУ им. М.В. Ломоносова в 1984 г. Д-р физ.-мат. наук, профессор кафедры «Физика» МГТУ им. Н.Э. Баумана. Автор 90 опубликованных работ. Области научных интересов: физика твердого тела, компьютерное моделирование, мессбауэровские и дифракционные исследования, магнитные материалы. e-mail: vintaik@mx.bmstu.ru, vintaikb@mail.ru