

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

3237/82

12/7-82

P11-82-205

В.И.Лумпов, Р.Н.Федорова

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЕ ВЕРСИИ
СИСТЕМЫ АНАЛИТИЧЕСКИХ ВЫЧИСЛЕНИЙ
REDUCE-2
ДЛЯ ЗАДАЧ ТЕОРЕТИЧЕСКОЙ МЕХАНИКИ

1982

Система REDUCE-2^{/1,2/} аналитических вычислений на ЭВМ является мощной универсальной системой аналитических вычислений /СAB/, которая с успехом может быть применена для ряда задач теоретической механики, связанных с большим объемом аналитических выкладок. Примером такой задачи может служить формализм уравнений Лагранжа 2-го рода. В отличие от системы^{/3/}, ориентированной на исследование устойчивости одного класса механических систем и являющейся специализированной системой, REDUCE-2 допускает решение более широкого класса задач. Однако в силу своей универсальности эта САВ оказывается менее эффективной и требующей для своей работы большого объема оперативной памяти ЭВМ. Особенно острой проблема памяти становится при работе САВ на ЭВМ с небольшим объемом памяти, например, младших моделях ЕС ЭВМ с памятью 512 К байт. Для того, чтобы эта САВ могла служить эффективным инструментом для исследования сложных механических систем, необходимо создание проблемно-ориентированных версий, более экономичных по памяти и в то же время достаточно эффективных.

В работе описывается методика и разрабатываемое математическое обеспечение создания проблемно-ориентированных версий /ПОВ/ САВ REDUCE-2 для ЕС ЭВМ.

Рассмотрим в качестве примера задачу вероятностного исследования нелинейных динамических систем на основе сведения ее к решению уравнения Фоккера-Планка-Колмогорова. Приближенные методы решения этого уравнения^{/4/} требуют выполнения достаточно громоздких символьных вычислений, включающих дифференцирование и интегрирование.

Поведение механической системы, подверженной действию случайных возмущений, может быть описано системой обыкновенных дифференциальных уравнений

$$\dot{x}_i + f_i(x_1, \dots, x_n, t) = \xi_i, \quad i=1, \dots, n, \quad /1/$$

где x_i - фазовые координаты, f_i - характеристики системы, предполагаемые достаточно гладкими в рассматриваемой области фазового пространства, ξ_i - случайные функции с матрицей спектральных плотностей

$$S = \{S_{ij}(t)\}, \quad i, j=1, \dots, n.$$

Для системы /1/ записывается уравнение относительно текущей плотности вероятности фазовых координат - уравнение Фоккера-

Планка-Колмогорова

$$\frac{\partial p}{\partial t} - \sum_{i=1}^n \frac{\partial}{\partial x_i} (p f_i) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n S_{ij} \frac{\partial^2 p}{\partial x_i \partial x_j} = 0. \quad /2/$$

Приближенное решение уравнения /2/ может быть, например, получено при помощи итерационно-операторного метода^{/5/} в виде

$$p_m = \sum_{i=0}^m B^i L^i p_0, \quad /3/$$

где L - линейный дифференциальный оператор,

$$L = - \sum_{i=1}^n \frac{\partial f_i}{\partial x_i} - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n S_{ij} \frac{\partial^2}{\partial x_i \partial x_j}, \quad /4/$$

$$B(u) = \int_0^t u dt, \quad /5/$$

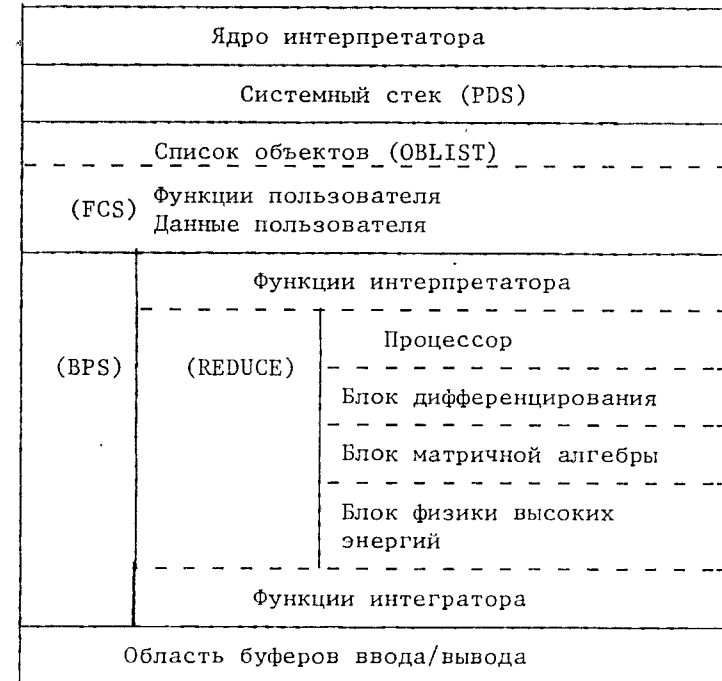
p_0 - плотность начального распределения.

Получаемое выражение для плотности используется с целью вычисления интегральных характеристик случайного процесса, описывающего поведение исследуемой системы, и обычно оформляется в виде подпрограммы-функции на фортране^{/6/}.

Для выяснения вопроса о возможностях экономии памяти во время выполнения программы в CAB REDUCE-2 рассмотрим ее структуру и использование оперативной памяти^{/7,8/}. Назовем стандартной системой вариант CAB, содержащий следующие компоненты /см. рисунок/:

- ядро интерпретатора, написанное на ассемблере и содержащее основные функции LISPA;
 - функции интерпретатора, не вошедшие в ядро, написанные на LISPe (FRONTEND), скомпилированные во время генерации встроенным компилятором и располагающиеся в BPS -участке памяти, отведенном для скомпилированных функций;
 - функции CAB, написанные на RLISPe, скомпилированные и также располагающиеся в BPS. Все множество этих функций разбито на блоки; работа некоторых из них не зависит от наличия или отсутствия ряда других. Примерами таких блоков могут служить блок работы с матрицами, блок физики высоких энергий и др.;
 - функции аналитического интегратора, также скомпилированные и располагающиеся в BPS;
 - функции пользователя, написанные на входном языке REDUCE-2, вычисляемые в режиме интерпретации. Эти функции располагаются в области списочной памяти - FCS.
- Компилятор в состав стандартной CAB не входит.

При решении задачи из некоторого класса не все функции, указанные выше, необходимы. Так, для реализации операторного метода не нужны блоки матричной арифметики, физики высоких



энергий, блок обеспечения работы CAB в интерактивном режиме при работе в пакетном режиме и др. Естественным поэтому является желание их удалить. Рассмотрим, из каких компонент и какими средствами ненужные блоки и отдельные функции могут быть удалены.

Ядро интерпретатора может содержать функции, не используемые в конкретном вычислении, например, функции для работы с разделами библиотечных наборов данных. Однако корректировка такой большой программы на ассемблере, как ядро интерпретатора, является задачей очень сложной, требующей детального знания как самого LISPA, так и интерпретатора, и далее не рассматривается.

Функции из FRONTEND могут быть легко удалены из исходного текста перед генерацией. Вопрос остается лишь в получении списка функций, подлежащих удалению.

Функции CAB могут быть удалены при генерации конкретного варианта CAB. Но с легкостью могут быть удалены лишь независимые блоки, что отмечено в комментариях к исходным текстам этих блоков. Что касается остальных функций, то выяснение возможности их исключения может быть осуществлено эксперименталь-

но. Прослеживание же вручную по исходному тексту всех связей между функциями крайне затруднено. Кроме рассмотренной, имеется также возможность удаления отдельных блоков при выполнении задач в САВ. Этот подход на примере блока физики высоких энергий описан в ^{9/}. К сожалению, он не универсален, так как с его помощью можно удалять функции и блоки из BPS, только начиная с некоторой функции и до конца этой области. Применительно к стандартной САВ, описанной выше, при помощи функции BPSWIPE можно удалить интегратор, или блок физики высоких энергий и интегратор, но нельзя удалить только блок матриц, оставляя интегратор. Освобождающаяся память включается в область списочной памяти FCS. Произвольное множество функций может быть удалено при генерации исключением их из исходного текста.

Из блока аналитического интегратора также может быть удален при генерации ряд функций, например, интегрирования рациональных функций, если они не используются в данном классе задач.

Таким образом, для удаления функций необходимо экспериментально получить список функций, вызывавшихся при решении задач данного класса. Для автоматизации получения списка используемых функций необходимо создание стартовой версии САВ, имеющей возможность отмечать вычисляемые функции и на основе этого корректировать исходные тексты блоков FRONTEND, REDUCE-2 и интегратора. Далее, по откорректированным исходным текстам проводится обычная генерация проблемно-ориентированной версии САВ.

Стартовая САВ отличается от стандартной модифицированной функцией EVAL. Эта функция включает в список свойств каждой вызываемой функции, исключая ядро интерпретатора, некоторый флаг, отмечающий использование ее в данном выполнении задачи. Пользователь, выполняя в стартовой САВ ряд задач из интересующего его класса, получает список использования функций, и на его основе - автоматически созданную ПОВ системы REDUCE-2. Следует отметить, что в исходных текстах содержится ряд функций, введенных временно, для целей раскрутки. Удаление их привело бы к невозможности генерации. Поэтому уже при создании стартовой САВ необходимо получить список таких функций и не удалять их из исходных текстов.

Перечислим компоненты, необходимые для автоматической генерации ПОВ:

- ядро интерпретатора с модифицированной EVAL;
- функция на LISPe, включенная в FRONTEND стартовой САВ для формирования набора данных, содержащего список используемых функций;
- функции на LISPe для корректировки исходных текстов FRONTEND, REDUCE и интегратора.

Тогда для генерации ПОВ необходимо выполнение последовательности заданий:

- в стартовой САВ решается достаточно представительный ряд задач заданного класса с накоплением списка использованных функций;
- выполняется задание на корректировку наборов для генерации САВ на основе полученного списка;
- генерация нового варианта LISPa с компилятором. Этот пункт может быть опущен и далее использован вариант из стандартной САВ;
- генерация нового варианта REDUCE-2 с компилятором;
- генерация нового варианта REDUCE-2 с интегратором и исключенным компилятором.

Полученная таким путем версия будет более эффективной за счет уменьшения числа сборок мусора, на ней можно будет решать более объемные задачи. На ряде задач сборка мусора может составлять до 30% времени. Таким представляется первый этап создания ПОВ. Здесь не рассматривались функции пользователя, написанные на входном языке САВ и вычисляемые в режиме интерпретации. При желании их можно скомпилировать и включить в BPS на этапе генерации интегратора. Не рассматривался также тривиальный подход, не связанный с экономией памяти, основанный на сохранении областей BPS и FCS во внешней памяти при помощи функции CHKPOINT^{10/}.

Недостатком рассмотренного подхода является то, что создание ПОВ требует не менее 600 К байт. Если же исключить этап создания интегратора, то ПОВ может быть получена с использованием лишь 400 К байт.

К настоящему времени разработаны функции на LISPe для формирования списка функций и корректировки исходных текстов, осуществлена модельная генерация с удалением блока матричной алгебры, разобрана реализация функции EVAL и осуществлены попытки ее модификации.

Дальнейшее развитие описанного подхода заключается в модификации функции EVAL и разработке дополнительных функций для работы с некомпилированными виртуальными функциями, т.е. функциями, которые могут вытесняться во внешнюю память и восстанавливаться в оперативной памяти при обращении к ним. Подобный подход реализован в LISP-системе на CDC-6500^{11/}. Однако полностью интерпретируемая САВ хотя и требует гораздо меньшей памяти, но работает существенно медленнее. Для достижения разумного компромисса между необходимым объемом оперативной памяти и эффективностью может быть реализован следующий подход.

На заданном классе задач собирается достаточно полная статистика использования функций. Тогда на основе собранной статистики все функции могут быть /в зависимости от числа обра-

щений, времени использования центрального процессора и объема необходимой для их представления памяти/ разделены на 2 класса: "редко" вызываемые и "часто" вызываемые. Возможны различные варианты формализации этих определений, оптимальность каждого из них нуждается в экспериментальной проверке. Тогда функции, отнесенные к "редко" используемым, могут быть оставлены при генерации в некомпилкованном виде. В список свойств таких функций вносится флаг VIRFN, по которому EVAL будет различать виртуальные функции.

В этом случае стартовая CAB должна иметь в своем составе два варианта ядра интерпретатора - одно для сбора и обработки статистики, и второе - для работы с виртуальными функциями. В качестве варианта возможно использование ядра с функцией EVAL, обладающей обоими свойствами. Тогда на конкретных задачах можно будет собирать статистику использования функций для уточнения определения "часто" и "редко" используемых функций. В данном варианте в ядро интерпретатора или в FRONTEND должны быть дополнительно включены функции:

- свертки определяющего выражения функции и развертки его при восстановлении в памяти;
- записи свернутого выражения во внешнюю память, в качестве которой естественно выбрать диски в силу достаточно интенсивного обмена в режиме произвольного доступа;
- восстановления выражений в оперативной памяти;
- выбора подлежащей вытеснению функции из числа виртуальных, находящихся в памяти.

Необходимо также внести изменения в определение EVAL в ветвях обработки EXPR и FEXPR и в сборщик мусора, который должен будет дополнительно в случае исчерпания памяти вытеснить виртуальные функции во внешнюю память. Подробное описание алгоритмов работы функции EVAL и сборщика мусора, а также дополнительных функций, требует значительного объема и здесь не рассматривается. К настоящему времени разобрана работа EVAL и сборщика мусора, определены ветви и структура вставок, разработан вариант программы свертки - развертки определяющих выражений, изучены средства и возможности прямого и библиотечного методов доступа в ОС ЕС.

Исходя из того факта, что алгоритм многих задач последовательно используют блоки системы, можно отметить возможность реализации подхода к созданию POB на основе своеобразного оверлея блоков в области BPS. В LISPe имеются функции для обеспечения такого перекрытия - BPSCHKPT и BPSRESTR. Реализация этого подхода не требует модификации ядра интерпретатора, не связана с корректировкой исходных текстов CAB для определения перекрывающихся сегментов. Управление перекрытием осуществляет пользователь во время выполнения его программы при помощи тех же функций.

В заключение следует отметить, что данное направление работ имеет целью создание проблемно-ориентированных версий CAB REDUCE-2, максимально экономичных по памяти в случае хорошо определенных и достаточно стабильных классов решаемых задач для применения на ЭВМ с небольшим объемом оперативной памяти. Поэтому результаты работ могут быть применены и в задачах, отличных от задач теоретической механики.

Авторы считают своим долгом выразить благодарность В.А.Ростовцеву за ряд полезных советов, в большой степени повлиявших на описанную выше идеологию создания проблемно-ориентированных версий.

ЛИТЕРАТУРА

1. Hearn A.C. REDUCE-2 User's Manual. Univ. of Utah, USA, 1973.
2. Hearn A.C. A New Release of REDUCE. REDUCE Newsletter, 1978, No.4.
3. Бурлакова Л.А., Иртегов В.Д., Почтаренко М.В. Использование символьных выкладок на ЭВМ в некоторых задачах механики. В сб.: Аналитические вычисления на ЭВМ и их применение в теоретической физике. Дубна, 1980, с.137-142.
4. Красовский А.А. Фазовое пространство и статистическая теория динамических систем. "Наука", М., 1974.
5. Маланин В.В. Механика, 1980, №1, с.А97.
6. Griss M.L. Symbolic-Numeric Tutorial. REDUCE Newsletter, 1979, No.7.
7. Marti J.B. et al. STANDARD LISP Report. Univ. of Utah, UCP-60, 1978.
8. Fitch J. Manual for STANDARD LISP on IBM System 360&370. Univ. of Utah, No.TR-6, 1975.
9. Hearn A.C. REDUCE-2 Implementation Guide for STANDARD LISP on the IBM System 360&370. Univ. of Utah, No.TR-5, 1979.
10. Hearn A.C. Global Saving of the REDUCE Environment. REDUCE Newsletter, 1979, No.7.
11. Kadantsev S.G., Rostovtsev V.A. Utilization of Secondary Memory in Running of the REDUCE-2. SIGSAM Bulletin, 1980, vol.14, No.4.

Рукопись поступила в издательский отдел
31 марта 1982 года.

НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

Вы можете получить по почте перечисленные ниже книги, если они не были заказаны ранее.

D1,2-9224	IV Международный семинар по проблемам физики высоких энергий. Дубна, 1975.	3 р. 60 к.
D-9920	Труды Международной конференции по избранным вопросам структуры ядра. Дубна, 1976.	3 р. 50 к.
D9-10500	Труды II Симпозиума по коллективным методам ускорения. Дубна, 1976.	2 р. 50 к.
D2-10533	Труды X Международной школы молодых ученых по физике высоких энергий. Баку, 1976.	3 р. 50 к.
D13-11182	Труды IX Международного симпозиума по ядерной электронике. Варна, 1977.	5 р. 00 к.
D17-11490	Труды Международного симпозиума по избранным проблемам статистической механики. Дубна, 1977.	6 р. 00 к.
D6-11574	Сборник аннотаций XV совещания по ядерной спектроскопии и теории ядра. Дубна, 1978.	2 р. 50 к.
D3-11787	Труды III Международной школы по нейтронной физике. Алушта, 1978.	3 р. 00 к.
D13-11807	Труды III Международного совещания по пропорциональным и дрейфовым камерам. Дубна, 1978.	6 р. 00 к.
	Труды VI Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1978 /2 тома/	7 р. 40 к.
D1,2-12036	Труды V Международного семинара по проблемам физики высоких энергий. Дубна, 1978	5 р. 00 к.
D1,2-12450	Труды XII Международной школы молодых ученых по физике высоких энергий. Приморско, НРБ, 1978.	3 р. 00 к.
	Труды VII Всесоюзного совещания по ускорителям заряженных частиц, Дубна, 1980 /2 тома/	8 р. 00 к.
D11-80-13	Труды рабочего совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике, Дубна, 1979	3 р. 50 к.
D4-80-271	Труды Международной конференции по проблемам нескольких тел в ядерной физике. Дубна, 1979.	3 р. 00 к.
D4-80-385	Труды Международной школы по структуре ядра. Алушта, 1980.	5 р. 00 к.
D2-81-543	Труды VI Международного совещания по проблемам квантовой теории поля. Алушта, 1981	2 р. 50 к.
D10,11-81-622	Труды Международного совещания по проблемам математического моделирования в ядерно-физических исследованиях. Дубна, 1980	2 р. 50 к.

Заказы на упомянутые книги могут быть направлены по адресу:
101000 Москва, Главпочтамт, п/я 79
Издательский отдел Объединенного института ядерных исследований

Лумпов В.И., Федорова Р.Н. P11-82-205
Проблемно-ориентированные версии системы аналитических вычислений REDUCE-2 для задач теоретической механики

Описываются методика и разрабатываемое математическое обеспечение создания проблемно-ориентированных версий системы аналитических вычислений REDUCE-2 для ЕС ЭВМ. Проблемно-ориентированная версия REDUCE-2 занимает меньший объем оперативной памяти, более эффективна и может быть применена для исследования сложных механических систем на ЕС ЭВМ с памятью 512 К байт. В качестве примера рассматривается задача вероятностного исследования нелинейных динамических систем на основе сведения ее к решению уравнения Фоккера-Планка-Колмогорова.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1982

Lumpov V.I., Fedorova R.N. P11-82-205
Problem Oriented Versions of REDUCE-2 System
of Analytical Calculations for Theoretical Mechanics Problems

Methods and software developed for problem-oriented versions of the REDUCE-2 system of analytical calculations for the ES computer are described. The REDUCE-2 version occupies the minor volume of core memory, is more effective and could be applied to investigate complicated mechanical problems on the ES computer with 512 K byte memory. As an example, the problem of probability investigation of nonlinear dynamic systems on the basis of its reducing to solution of the Fokker-Planck-Kolmogorov problem is considered.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1982

Перевод О.С.Виноградовой.