ных значения; род/число – 3 значения (мужской и женский род, множественное число); 3 лица; 5 типов артиклей; 6 видо-временных форм глагола. На первый взгляд может показаться странным очень большое количество частей речи и небольшое количество остальных характеристик, однако это вполне объяснимо характером создаваемой модели. Характеристики, уточняющие части речи на уровнях «словоформа – сегмент – предложение» являются согласующими признаками. В максимальных формах для переходов между этими уровнями используется характеристика части речи, а её уточнения описываются как параметры, требующие или не требующие согласования внутри объекта, описываемого максимальной формой.

Из вышеизложенного становится понятным, почему среди 53 частей речи - 6 различных прилагательных, 18 местоимений, 9 глаголов, 6 причастий.

Сегменты предложений (как самостоятельные, так и в виде рекурсивных вставок в другие сегменты) описываются 27 максимальными формами. Приведем пример описания максимальной формы одного из сегментов, а именно числительного (C):

$$C = (IN)(IN)(PI)(PI'')(A)(IN)\ C\ (P'')(J)(J'')(IN)$$
$$0\ 0\ 0\ 0\ 2\ 0\ 2\ 2\ 2\ 2\ 0$$

Ряд чисел под последовательностью частей речи означает необходимость их согласования по некоторым характеристикам, в данном случае – по роду/числу в позиционированном представлении лексико-грамматических классов слов [3]. В скобках указаны необязательные компоненты формы, т.е. в конкретных реализациях они могут отсутствовать.

Среди максимальных форм выделены одна номинативная (существительного), 10 местоименных, 1 - числительного, по 3 причастных и деепричастных, 9 глагольных форм.

Анализ целесообразности использования предложенного распределения лингвистической информации между понятиями модели «часть речи» и «уточняющие характеристики» приводит к выводу, что существуют закономерности в текстах любого естественного языка, которые можно описать простой линейной зависимостью. Простота этой зависимости проявляется только после более широкого толкования рассматриваемых лингвистических характеристик по отношению к классическому их определению в лингвистике.

Очевидно, что полноценное представление всех свойств текста в рамках линейной модели невозможно. Однако даже легко формализуемые закономерности в рамках этой линейной модели требуют достаточно существенных усилий по приведению общепризнанных лингвистических характеристик к несколько искусственной группировке. Это, по всей видимости, относится к любым попыткам построить реляционную модель текста, пусть это будет статистическая модель типа марковской либо иерархическая, кажущаяся многоуровневой, однако таковой по сути не являющаяся [6,7]. Моделью, адекватной человеческому восприятию, может быть лишь синтетическая модель, состоящая из набора различных моделей реляционного и сущностного характера, сочетание которых определяется только в зависимости от конкретной проблемы, решаемой при анализе того или иного фрагмента текста. С этой точки зрения совокупные категориальные характеристики, как оконечные признаки, описывающие ту или иную единицу текста, можно считать достаточно универсальным средством представления такого описания. Именно поэтому их используют в различных статистических моделях, не учитывающих структуру текста.

Заключение

В целом иерархическую модель можно считать некоторым «скелетом», на который, в зависимости от решаемой задачи, тем или иным образом наращиваются «мягкие ткани», представляющие собой различное отражение классических лингвистических понятий. Структурирование линейного представления лингвистической информации в виде совокупных категориальных характеристик, введение новых классов, представляющих уже известные объекты, помогают учитывать закономерности, присущие тексту с точки зрения цели конкретной задачи.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Antonov S., Nikishin A. Elements of Hierarchic Model of a Text. 12th World Congress of Applied Linguistics. AILA'99 Tokyo, August 1-6, 1999, Program &Abstracts p.389.
2. Антонов С.Г., Никишин А.Г. Метод построения иерархической модели текста. // Материалы международной конференции «Когнитивное моделирование в лингвистике», Переславль-Залесский, 2000. / Обработка текста и когнитивные технологии, №5 – М., 2001, с. 35-42.
3. Антонов С.Г. Позиционная модель именного сегмента в тексте немецкого языка. Вестник Минского государственного лингвистического университета. Серия 1. Филология. №14 – Мн.: МГЛУ. – 2004, с.158-162.
4. Яковишин В.С. Формальный язык. Теория. Грамматика. Применение. – Минск: НАНРБ, 2000.
5. Антонов С.Г., Никишин А.Г., Пересыпкин В.А., Степанов А.В. Об информационных свойствах морфологической модели текста. Актуальные проблемы компьютерной лингвистики. Сб. научных статей. – Мн., 2005. – С.16-24.
6. Зубов А.В. Статистика и моделирование как основа систем искусственного интеллекта // Международный семинар "Язык и технология" – Санкт-Петербург, 1996, с.16.
7. Антонов С.Г., Богушевич Д.Г., Зубов А.В., Нехай О.А., Никишин А.Г. К проблеме построения формальных моделей естественных языков. Вестник Минского государственного лингвистического университета. Серия 1. Филология. №4 – Мн.: МГЛУ. – 1998, с.165-169.

Статья поступила в редакцию 21.05.2007

УДК 004.421.2

Афонин В.Г.

О НЕКОТОРЫХ ПРОЦЕДУРАХ ВЫЧИСЛИТЕЛЬНОЙ ПРАКТИКИ

Введение. Важнейшим итогом решения любой вычислительной задачи является степень доверия к полученным результатам, оценка их погрешности и устойчивость вычислительного процесса. Эта устойчивость (корректность) является также важнейшей характеристикой всей математической модели реальной задачи. В п. 1.1 данной работы предлагается весьма простой и высокоэффективный приём установления степени корректности широкого класса вычислительных алгоритмов.

Как правило, в математических моделях присутствуют параметры, которые получаются в результате измерений. Эти измерения содержат погрешности, зачастую немалые. Кроме

Афонин Владимир Гаврилович, *к.ф.-м.н., доцент кафедры информатики и прикладной математики БрГТУ.*
Беларусь, Брестский государственный технический университет, 224017, г. Брест, ул. Московская 267.

того, некоторые параметры задачи могут управляемо меняться, и требуется исследовать влияние этих изменений. В связи с этим возникает общая задача проверки зависимости изменения выходных параметров от меняющихся входных. Один из возможных способов решения этой задачи предлагается в п. 1.2.

Наконец, важнейшим этапом компьютерного решение любой вычислительной задачи является отладка программы (вычислительного документа). Этому вопросу посвящён п.2 настоящей работы, где предлагается простая и эффективная технология конструирования «отладочной» задачи, близкой к исходной.

1. Пусть при численном решении некоторой задачи по входным параметрам X_i, $i=1,2,\ldots,m$ вычисляются выходные параметры Y_j, $j=1,2,\ldots,n$.

Будем предполагать, что все величины имеют вещественный тип (это чаще всего выполняется на практике) и вычисления ведутся в режиме с плавающей точкой.

1.1. Для проверки устойчивости конкретного вычислительного процесса к погрешностям округления предлагается решать данную задачу (с одними и теми же исходными данными) по крайней мере, дважды, причем с разным количеством значащих цифр. Например, с одинарной точностью (single) и с удвоенной точностью (double) - это достаточно просто делается в разных средах программирования.

Далее полученные решения сравниваются между собой, и по величине отклонения делаются соответствующие заключения. Если отклонения незначительны, вычислительный процесс (при конкретных исходных данных) считается устойчивым к вычислительным погрешностям.

В случае значительного расхождения полученных решений вычислительный процесс считается неустойчивым (задача - некорректной), и степень доверия к полученным результатам снижается. Например, легко и просто, проводя вычисления с одинарной и удвоенной точностью, можно убедиться в некорректности решения задач численного дифференцирования с малым шагом, где производится вычитание близких чисел.

Но, после установления некорректности, возникает проблема обнаружения конкретных источников вычислительной погрешности. Ясно, что эта задача во многих случаях является достаточно сложной.

Для ее решения предлагается использовать следующую процедуру.

Хорошо известно, что наибольшую погрешность (точнее, наибольшую потерю значащих цифр) дает операция вычитания двух близких чисел, у которых значительная часть цифр старших разрядов совпадает. При этом, вопреки распространенному мнению, эти числа не обязательно должны быть "большими". Ведь при машинных вычислениях с плавающей точкой значащие цифры определяются мантиссой, а сама величина числа "уходит" в порядок, от которого точность вычислений в принципе не зависит.

Пусть, например, при работе с 6 верными значащими цифрами (single) произошло совпадение 3-х старших разрядов операндов операции вычитания. Это приводит к результату, содержащему всего 3 верные значащие цифры. Еще одно такое вычитание операндов, хотя бы один из которых имеет только 3 верные значащие цифры - и все значащие цифры результата будут неверными.

При этом необходимо отметить, что каждое "дефектное" число, содержащее, например, лишь 3 верные значащие цифры вместо 6, приводит к потере значащих цифр результатов выполнения обычных операций, в которых это число участвует. То есть, происходит как бы "заражение" потерей точности.

Результат, аналогичный только что рассмотренному вычитанию, дает операция сложения двух близких по модулю чисел с противоположными знаками.

Для достаточно точного обнаружения таких операций в исходном тексте программы предлагается сформировать ее контрольный вариант, в котором следует заменить выражения типа $a+b$ и $a-b$ на соответствующие процедуры, которые будут содержать операнды a и b в качестве фактических параметров. Кроме того, фактическими параметрами здесь могут служить порядковый номер строки исходного текста и порядковый номер операции сложения и вычитания в этой строке. В результате каждый вызов такой процедуры позволит получить информацию о том, какие операции (и, если они выполняются многократно, на каком именно шаге) дают наиболее значительные погрешности. Такая информация позволит пользователю внести необходимые коррективы в исходный текст и в определенных случаях избежать значительной потери точности, то есть существенно улучшить вычислительные свойства алгоритма, сделать его более устойчивым.

Предлагаемые процедуры можно сформировать на обычных алгоритмических языках программирования высокого уровня (Fortran, Basic, Pascal, C).

1.2. Для проверки зависимости изменения выходных параметров от меняющихся входных предлагается следующая достаточно естественная процедура.

Пусть каждая из исходных величин X_i может меняться в определенном диапазоне и пусть она принимает N_i значений из этого диапазона.

Будем предполагать, что величины X_i меняются независимо друг от друга. Тогда общее количество точек $\{X_1, X_2, \ldots, X_m\}$ - узлов пространственной сетки - m - мерного параллелепипеда Ω составит $N_1 \cdot N_2 \cdot \ldots \cdot N_m$.

Решая задачу отыскания выходных параметров Y_j для каждого из этих узлов, можно провести достаточно полный численный анализ зависимости результатов вычислений от исходных данных.

Основным недостатком такого подхода является весьма значительный объем вычислений.

Но есть и существенное достоинство: возможность естественного распараллеливания вычислительного процесса. Именно, параллелепипед Ω можно разделить на несколько блоков и работать с ними по отдельности, задействовав вычислительные возможности нескольких процессоров. И здесь высокий эффект может дать параллельное использование нескольких компьютеров (компьютерных сетей), а также суперкомпьютеров (типа СКИФ, например).

Если входные параметры взаимосвязаны (не являются независимыми), область их изменения имеет более сложную форму, чем параллелепипед. Но и в этом случае естественное распараллеливание имеет место.

Предлагаемый весьма простой, но практически важный прием можно использовать даже в обычных учебных курсах, в том числе и для демонстрации одного из направлений эффективного применения распараллеливания вычислительных процессов.

В случае, если данная задача представляет собой математическую модель какого-либо объекта, вычислительные эксперименты такого рода позволят получить достаточно полное представление о возможных вариантах поведении этого реального объекта.

При этом желательно проводить и процедуры, описанные в п.п. 1.1. Ведь вычислительная неустойчивость при определенных значениях входных параметров $X_1, X_2, \ldots, X_m$ может служить признаком неустойчивости самого реального объекта (моделируемого процесса, явления), то есть, оказаться практически полезным.

Таким образом, предлагаемый универсальный подход к решению многих вычислительных задач позволяет провести достаточно глубокий анализ процесса вычислений без аналитических выкладок, что во многих случаях (особенно когда аналитическое исследование затруднено или вообще невозможно) имеет определенную практическую ценность.

2. Ниже предлагается еще один эффективный методологический прием вычислительной практики, который автору не удалось найти ни в учебной литературе, ни в каких-либо других источниках информации.

Пусть нам нужно, к примеру, получить решение какой-либо задачи для обыкновенного дифференциального уравнения (ОДУ) вида

$$F(x, y, y', y'', \ldots, y^{(n)})=0, \quad (1)$$

где $y=y(x)$ - неизвестная функция, удовлетворяющая каким-либо граничным, либо начальным условиям, либо другим условиям единственности решения.

Для апробации и отладки алгоритма предлагается решить задачу с заранее известным ответом, полагая $y=p(x)$, где *известная* функция $p(x)$ удовлетворяет, вообще говоря, другим условиям единственности. То есть, будет решаться задача для ОДУ вида

$$F(x, y, y', y'', \ldots, y^{(n)}) = F(x, p, p', p'', \ldots, p^{(n)}), \quad (2)$$

которая имеет довольно много общего с (1). При этом отрезок, где ищется решение задачи (2) можно взять небольшим для получения достаточно высокой точности при минимальном объеме вычислений.

После такой успешной отладочной апробации можно применять алгоритм к решению исходной задачи (1).

Значительный эффект от такого подхода можно получить, если используются типовые методы для решения краевых задач:

- метод конечных разностей (метод сеток),
- методы типа Бубнова – Галеркина или Рэлея – Ритца,
- методы конечных элементов, которые можно рассматривать как метод Ритца со специфическим выбором базисных функций,

а также другие методы, где решающую роль играет левая часть уравнения (2).

Очевидно, подобный прием можно успешно применять при решении самых разнообразных конечных, дифференциальных и интегральных уравнений и их систем, а также для решения других задач вычислительного характера.

Например, решая n – систему конечных уравнений вида

$$f(x) = 0, \quad (3)$$

можно вначале решить одну или несколько систем вида

$$f(x) = f(x_p), \quad (4)$$

где x_p - известный "пробный" вектор, и только затем уже решать систему (3).

При этом, вообще говоря, желательно, чтобы "пробный" (отладочный) вектор был по возможности близок к искомому решению. Это требование может быть реализовано на практике, когда приближенное решение данной задачи известно, например, из физических соображений.

Ввиду простоты и эффективности предлагаемых процедур, их можно рекомендовать для включения в разнообразные учебные курсы для образовательных заведений.

Заключение. Предлагаемые приёмы вычислительной практики достаточно просты в реализации и могут эффективно использоваться при решении самых разнообразных задач вычислительного характера. В идеале такими приёмами могут владеть не только специалисты в области вычислительной математики, но и инженеры, и экономисты, а также студенты соответствующих специальностей.

Статья поступила в редакцию 15.03.2007

УДК 741.02:519

Игнатюк В.И., Игнатов А.Ю.

МОДЕЛИРОВАНИЕ ВРАЩЕНИЯ АКСОНОМЕТРИЧЕСКОГО ИЗОБРАЖЕНИЯ ПРОСТРАНСТВЕННОЙ СТЕРЖНЕВОЙ СИСТЕМЫ НА ЭКРАНЕ МОНИТОРА

Введение. Изображение пространственной системы на экране монитора, представляющем собой по существу плоскость, требует преобразования ее в плоское изображение, как можно более наглядно отображающие исходную пространственную систему. Такое преобразование может быть выполнено с использованием, так называемых, аксонометрических проекций [1]. При этом плоскость, на которую проецируется изображение (плоскость экрана монитора), новые оси в этой плоскости, проекции точек и элементов также называют аксонометрическими. При таком преобразовании пространственной системы в плоское изображение, естественно, возникают искажения ее вида, которые описываются с помощью коэффициентов искажения k_x, k_y, k_z, представляющих собой отношения аксонометрических координат точек изображения на соответствующих осях (или длин соответствующих отрезков) к действительным координатам этих точек (длин отрезков).

Следует иметь в виду, что аксонометрические проекции могут иметь [1] самые разные виды. Они могут быть изометрическими (при одинаковых коэффициентах искажения по всем осям), диметрическими (если коэффициенты искажения равны только по двум осям) и триметрическими (если по всем осям коэффициенты искажения различны). С другой стороны, изометрия, диметрия и триметрия могут быть прямоугольными и косоугольными. Системы декартовых координат могут быть правыми и левыми.

Выбор аксонометрии. В качестве пространственных объектов в данной работе рассматриваются пространственные стержневые системы, расчетные схемы которых [2] состоят из узлов (точек) и стержней, присоединяющихся к этим узлам (по существу, соединяющих указанные точки). Анализ представления таких систем, выполненный при рассмотрении вопроса, с учетом также особенностей представления аналитических зависимостей, которые будут применяться для расчета систем [2, 3], привел к решению использовать для представления рассматриваемых пространственных стержневых систем правую аксонометрическую прямоугольную диметрию в виде, представленном на рис. 1. Здесь θ_y – угол направления оси y в аксонометрии (по отношению к оси x), принятый равным 35°.

Аксонометрические координаты точек системы через их действительные координаты выражаются следующим образом:

Игнатюк Валерий Иванович*, к.т.н., доцент, заведующий кафедрой строительной механики БрГТУ.*
Игнатов Алексей Юрьевич, *студент строительного факультета БрГТУ.*
Беларусь, Брестский государственный технический университет, 224017, г. Брест, ул. Московская 267.