

В. Н. Гридин¹, науч. руководитель, д-р техн. наук, проф., e-mail: info@ditc.ras.ru,

В. И. Анисимов^{1,2}, гл. науч. сотр., д-р техн. наук, проф., e-mail: info@ditc.ras.ru,

М. М. Абухазим², аспирант, e-mail: Abuhazim_monther@Yahoo.com,

¹ Центр информационных технологий в проектировании РАН,

² Санкт-Петербургский государственный электротехнический университет

Индексно-численная технология сжатия данных в системах автоматизации схемотехнического проектирования*

Рассматриваются способы увеличения производительности систем автоматизированного проектирования на основе перехода к компактной форме хранения и обработки разреженных матриц. Дается описание наиболее эффективного метода компактного хранения и обработки разреженных матриц на основе строчно-столбцового фиксированного формата. Показывается, что отличительной особенностью методов фиксированного формата является невозможность включения в описание дополнительных ненулевых элементов, что исключает возможность непосредственного применения этих методов для обработки информации при решении систем уравнений любым численным методом, вследствие неизбежного появления новых ненулевых элементов в процессе этого решения. Для возможности учета новых ненулевых элементов в компактном описании моделируемой системы предлагается использовать индексно-численную технологию формирования такого описания и дается описание реализации сжатия данных на основе разделения общей процедуры на две независимые части индексного и численного анализа. Приводится методика построения программного обеспечения систем автоматизации схемотехнического проектирования на основе индексно-численной технологии сжатия данных.

Ключевые слова: системы автоматизированного проектирования, моделирование систем, компактная обработка, разреженные матрицы, строчно-столбцовый фиксированный формат

Введение

Наличие в математическом описании моделируемых систем разреженных матриц требует изменения стандартных подходов к формированию и решению систем уравнений, что вызвано, с одной стороны, необходимостью экономии памяти, которую нежелательно использовать для хранения нулевых элементов, а с другой стороны, необходимостью повышения быстродействия за счет исключения арифметических операций с нулевыми элементами. Эта задача особенно актуальна при построении систем с распределенной архитектурой, когда информационные ресурсы предоставляются потребителям посредством сетевых сервисов [1–4]. Применяемый обычно способ логической проверки элементов в целях устра-

нения арифметических операций с нулевыми элементами не приводит к ожидаемому эффекту увеличения быстродействия программного обеспечения, поскольку для выполнения логических операций проверки элементов также необходимо затратить определенное время.

Известные методы компактной обработки разреженных матриц существенно отличаются по своим характеристикам и своей эффективностью [5–9], при этом наиболее высокими показателями степени экономии памяти характеризуется метод строчно-столбцового фиксированного формата. Однако этот метод вследствие фиксированности формата не позволяет вводить в ходе обработки дополнительные элементы в описание. Вместе с тем новые ненулевые элементы неизбежно появляются в процессе решения уравнений моделируемой схемы. Поскольку формат описания в фиксированном формате жестко задан, то в нем не зарезервированы места для размещения новых

*Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта 18-07-00082а.

ненулевых элементов, что ограничивает возможности метода в классической форме только решением задач хранения в сжатой форме неизменной информации.

В настоящей статье рассматривается индексно-численная технология сжатия данных, основанная на использовании двухэтапной процедуры построения компактного описания, включающей в себя индексный и численный этапы. При этом на индексном этапе уточняется формат описания схемы с учетом возможного появления новых ненулевых элементов в процессе решения уравнений схемы, а на численном этапе в уточненном формате формируется компактное описание моделируемой схемы без предварительного построения ее полных разреженных матриц. На заключительном этапе выполняется решение уравнений моделируемой схемы, при этом расчет переменных осуществляется виртуальными алгоритмами на основе компактных численных массивов без отображения в памяти компьютера полной разреженной матрицы.

Индексно-численная технология сжатия данных на основе строчно-столбцового фиксированного формата

Метод строчно-столбцового формата [5, 6] является наиболее эффективным методом сжатия данных и основан на использовании фиксированного формата, при этом предполагается, что исходная матрица W является структурно-симметричной, такой что каждому ненулевому элементу w_{ij} можно поставить в соответствие элемент w_{ji} . В случае если такой элемент в исходной матрице отсутствует, то его необходимо создать искусственно путем включения в компактное описание элемента $w_{ji} = 0$.

Для компактного хранения исходных элементов матрицы, требуется создать три массива:

WD — для хранения диагональных элементов;

WL — для хранения ненулевых элементов, расположенных ниже диагонали (поддиагональных элементов);

WU — для хранения ненулевых элементов, расположенных выше диагонали (наддиагональных элементов).

В соответствии с методом при формировании массива WU наддиагональные элементы записываются по строкам, а при формировании массива WL поддиагональные элементы записываются по столбцам. Согласно принятому порядку формирования массивов WU , WL

относительный адрес некоторого элемента w_{ij} , расположенного в массиве WU , совпадает с относительным адресом элемента w_{ji} , расположенного в массиве WL . Отмеченное свойство существенно упрощает процесс программирования для организации сканирования элемента.

Для хранения индексов строк и столбцов ненулевых элементов в методе строчно-столбцового фиксированного формата используется массив WJI , при этом этот массив содержит номера столбцов ненулевых элементов, расположенных выше диагонали, которые совпадают с номерами строк транспонированных ненулевых элементов, расположенных ниже диагонали. Для определения точки входа в строку выше диагонали (точка входа в столбце ниже диагонали) используется массив ERC . В последний элемент этого массива заносится значение $(m - n)/2 + 1$.

Степень эффективности метода может быть определена коэффициентом эффективности использования памяти $\beta = M/M_1$, где M — объем памяти, требуемый для полного описания разреженной матрицы, M_1 — объем памяти, требуемый для компактного описания разреженной матрицы.

Если учесть, что длина массива WD составляет n элементов, длина массивов WU , WL , WJI составляет $(m-n)/2$ элементов, а длина массива ERC составляет n элементов, то эффективность метода строчно-столбцового фиксированного формата может быть определена выражением

$$\beta = \frac{8n^2}{n^2\alpha \cdot 9 + n}.$$

Отсюда следует, что эффективность метода приближается к максимально-возможному значению $1/\alpha$ и дальнейшее снижение объема памяти при компактном описании уже практически невозможно.

Поскольку формат всех массивов жестко зафиксирован и не должен меняться произвольным образом в процессе расчета, отличительной особенностью метода является невозможность включения в массивы дополнительных ненулевых элементов. Это обстоятельство исключает возможность непосредственного применения метода строчно-столбцового фиксированного формата для решения уравнений любым численным методом вследствие неизбежного появления новых элементов с ненулевыми значениями при решении уравнений. Поэтому в классической форме метод строчно-

столбцового фиксированного формата может быть практически использован только для хранения разреженных матриц в сжатой форме.

Следовательно, для возможности полноценного использования метода строчно-столбцового формата на всех шагах расчета необходима его модификация путем введения двухэтапной процедуры построения компактного описания, включающей в себя индексный этап и численный этап [10]. На первом, индексном этапе определяются только размеры всех массивов с учетом появления новых элементов с ненулевыми значениями. Поскольку на этом этапе не ставится задача формирования и обработки численной информации, то его выполнение осуществляется путем обработки некоторой целочисленной матрицы C , элементы которой имеют только два произвольных значения, например 0 и 1. Для формирования численных массивов служит второй этап, где используются определенные на первом этапе форматы массивов, учитывающие зарезервированные места для всех вновь созданных ненулевых элементов массивов. Процесс индексно-численной технологии сжатия данных может быть реализован согласно блок-схеме, приведенной на рис. 1.

Согласно блок-схеме на этапе индексного анализа (шаги 1–3) формируется индексная матрица C , имеющая такие же размеры, как исходная матрица. На следующем шаге индексного этапа проводится LU -факторизация индексной матрицы, целью которой является уточнение формата компактного описания. В процессе индексной LU -факторизации выполняется упорядочивание строк и столбцов индексной матрицы в целях обеспечения минимального числа вновь появившихся ненулевых элементов. На заключительном третьем шаге индексного этапа осуществляется построение координатных матриц ERC , WJI , SI , в которых будет содержаться информация о структуре матрицы C , и, следовательно, матрицу C можно удалить.

На численном этапе (шаги 4–10) на основании уже известного формата описания открываются массивы, где будут формироваться численные матрицы WU , WL , WD , и выполняется формирование соответствующих массивов. Здесь же приводится численная LU -факторизация, при этом LU -факторизация выполняется виртуально путем обработки компактных массивов WU , WL , WD , так как полная разреженная матрица не существует, и необходима разработка специального алгоритма LU -факторизации, отличающегося от известных алгоритмов общего вида.

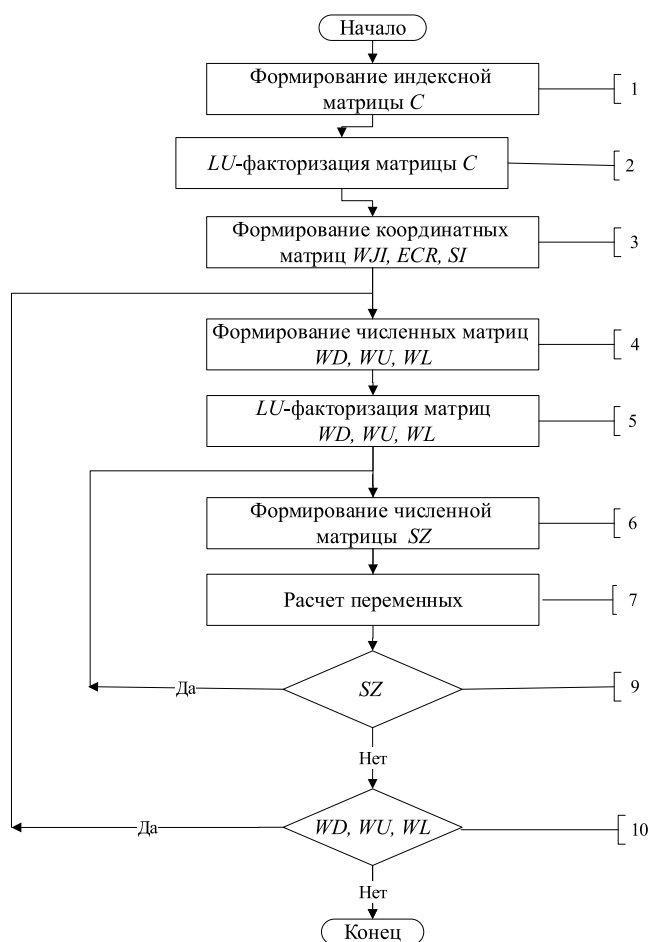


Рис. 1. Блок-схема индексно-численной технологии сжатия данных

После завершения численного этапа выполнения LU -факторизации на шаге 6 осуществляется формирование массива задающих источников SZ , и на шаге 7 выполняется расчет всех переменных системы и перекодировка этих переменных согласно проведенному на этапе индексного анализа упорядочиванию столбцов и строк. Такая задача согласно общей методике LU -факторизации решается с помощью обратного и прямого хода по строкам исходной матрицы. Так как исходная матрица отсутствует, то прямой и обратный ход следует выполнять на основании компактного описания, что также требует создания специальных алгоритмов.

На шаге 9 осуществляется проверка необходимости изменения содержимого массива SZ задающих источников и, если необходимо, выполняется возврат к шагу 6. На шаге 10 осуществляется проверка необходимости изменения содержимого массивов численных параметров схемы, содержащихся в компактных массивах WU , WL , WD , и, если необходимо, выполняется возврат к шагу 4.

Существенным достоинством рассмотренной индексно-численной процедуры является разделение ее на две независимые части, содержащие индексный и численный этапы. Так как практически все реальные задачи разработки электронных схем связаны с многовариантным расчетом схемы выбранной структуры, то индексный этап выполняется для каждой структуры единственный раз, тогда как численный этап реализуется десятки, сотни а иногда и тысячи раз.

Однако индексно-численная процедура характеризуется довольно сложной логикой разработки программного обеспечения, и при построении системы моделирования на основе компактного описания целесообразно использовать созданное ранее полное математическое описание задачи в качестве прототипа для построения ее компактного описания.

Модификация программного обеспечения на этапе индексного анализа

Под модификацией программного обеспечения будем понимать переработку существующего программного обеспечения, основанного на полном математическом описании задачи, в целях перехода к новому описанию моделируемой схемы и решению сформированных уравнений на основе компактного описания задачи согласно блок-схеме, представленной на рис. 1. Наличие прототипа значительно упрощает процесс создания программного обеспечения, основанного на сжатии данных, однако для наиболее эффективного использования имеющегося материала необходимо иметь в распоряжении методику разработки модифицированных версий программного обеспечения.

На первом шаге этапа индексного анализа выполняется формирование индексной матрицы, для которой необходимо создать соответствующий массив размерности $(n + 1)(n + 2)$, где n — общее число переменных для текущей задачи с учетом расширения базиса. Основная информация о включении компонентов схемы хранится в строках и столбцах, индексы которых находятся в интервале от 1 до n , в нулевом столбце массива хранится информация о месторасположении задающих источников, а в нулевой строке содержится вектор P , в котором хранится информация о всех перестановках строк и столбцов матрицы схемы. Столбец с индексом $n + 2$ также является вспомогательным и в нем хранится информация о числе

наддиагональных ненулевых элементов, которая используется при работе алгоритма минимальной степени (алгоритма Марковица) при выборе оптимальной последовательности строк (столбцов), обеспечивающей минимизацию числа новых значащих элементов при выполнении LU -преобразования.

Для формирования индексной матрицы при решении задачи модификации программного обеспечения на основе имеющегося прототипа следует использовать имеющиеся в этом прототипе функции формирования описания компонентов схемы `form_str` и построить на их основе новую совокупность функций `sform_str`, единственное отличие которых от прототипа `form_str` заключается в замене оператора формирования каждого ненулевого элемента массива $w[i][j]$, отображающего разреженную матрицу схемы, оператором формирования соответствующего элемента индексной матрицы $C_{ij} = 1$. Для завершения формирования массива C следует заполнить его нулевую строку и $(n + 2)$ -й столбец, где хранится информация о числе наддиагональных ненулевых элементов, для чего удобно создать вспомогательную функцию.

Для выполнения индексного этапа LU -факторизации необходимо построить вспомогательную функцию, осуществляющую замену процедуры реальной LU -факторизации ее эквивалентом на основе "портрета" схемы, отображенной в индексной матрице. Обработка "портрета" схемы должна выполняться одновременно с проведением оптимального упорядочивания нумерации строк (столбцов) матрицы C . Для такого упорядочивания наиболее эффективным является алгоритм минимальной степени [11], который реализуется исходя из выбора минимального значения произведения $r_k c_k$, где r_k — число ненулевых элементов в k -й строке на k -м шаге, а c_k — число ненулевых элементов в k -м столбце. Применительно к структурно-симметричной матрице выполняется соотношение $r_k = c_k$, и алгоритм минимальной степени приводит к требованию минимизации числа элементов в k -й строке. При этом для сохранения структурной симметрии в процессе оптимального упорядочивания необходимо осуществлять перестановку как строк, так и столбцов. Поскольку перестановка столбцов приводит к перестановке вектора переменных, то информацию о выполненных перестановках необходимо обязательно сохранять до конца расчета, для чего используется нулевая строка массива C .

Помимо решения задачи оптимального упорядочивания строк и столбцов на основе критерия минимизации появления новых значащих элементов при выборе ведущего элемента на этапе индексного анализа процесса LU -факторизации следует также обеспечить ненулевое значение этого элемента в полном описании матрицы W . Очевидно, что только такое условие позволит выполнить пересчет элементов на численном этапе LU -факторизации. Вместе с тем при описании систем в расширенном базисе узловых потенциалов [12, 13] не равные нулю значения диагональных коэффициентов матрицы обеспечиваются только для значений n , не превышающих величины nv , где nv — число узлов схемы. При $n > nv$ диагональные элементы могут иметь нулевые значения, и, следовательно, выбор их в качестве ведущих недопустим. Чтобы устранить указанное препятствие, следует соответствующим образом организовать работу оптимального упорядочивания строк и столбцов в процессе индексной LU -факторизации. С этой целью в начале упорядочивания следует провести перестановки строк (столбцов), индексы которых не превышают значение nv , а затем осуществить процедуру упорядочивания строк и столбцов, изменяя индексы в пределах от $nv + 1$ до n . Если следовать указанному алгоритму оптимального упорядочивания, то в процессе пересчета элементов при $n \leq nv$ все диагональные элементы для $n > nv$ приобретут ненулевые значения, и работа на численном этапе окажется возможной.

Таким образом, процедура LU -факторизации на этапе индексного анализа должна выполняться в следующей последовательности.

1. Ввести вспомогательный вектор N , начальные значения элементов которого определяются числом ненулевых недиагональных элементов в i -й строке матрицы C , а также вектор перестановок P , начальные значения элементов которого определяются натуральным рядом целых чисел.

2. Среди значений N_i ($i = \overline{1, nv}$) выбрать элемент N_i с минимальным значением и осуществить перестановку i -й и l -й строк и i -го и l -го столбцов, а также i -го и l -го элементов векторов N и P .

3. Выполнить первый шаг LU -факторизации матрицы C . Если $C_{i1} \neq 0$ и $C_{li} \neq 0$, а $C_{ij} = 0$, то в матрицу C следует ввести элемент $C_{ij} = 1$.

4. Провести модификацию вектора N путем прибавления к элементу N_i единицы с проявлением каждого нового ненулевого элемента и

вычитания единицы из каждого элемента N_j ($j = \overline{2, n}$), для которого $C_{j1} \neq 0$. При этом модифицированный вектор N будет характеризовать насыщенность строк подматрицы $[C]_2^n$ ненулевыми недиагональными элементами.

5. Выполнять шаги 2—4, пока $k \leq nv$, после чего искать $N_i = N_{i_{\min}}$ среди строк с индексами $i = nv, n$.

После проведения индексной LU -факторизации необходимо провести подсчет числа ненулевых наддиагональных элементов ni в индексной матрице на основании информации, содержащейся в $(n + 2)$ -м столбце массива C , а также подсчет числа значащих элементов ns в его нулевом столбце, для чего удобно построить вспомогательную функцию (например, us). Определенное в результате работы этой функции значение ni определяет формат массива наддиагональных элементов WU и массива поддиагональных элементов WL , а значения ns — формат массива SI .

Для формирования координатного описания целесообразно построить вспомогательную функцию (например, $cord$), целью которой является построение массивов WJI , ERC , SI , отображающих соответствующие матрицы, формат которых уже определен на предыдущих шагах индексного этапа. Координатные массивы WJI , ERC , SI , а также массив перестановок P должны быть открыты после окончания работы функции us , реализующей расчет форматов массивов. Компактный массив WJI содержит индексы столбцов исходной полной матрицы в ее наддиагональной части и вследствие структурной симметрии исходной матрицы одновременно содержит также и индексы столбцов ненулевых элементов в ее поддиагональной части. Связанный с индексом j -го столбца относительный адрес aw может быть найден путем последовательного сканирования строк индексной матрицы C с подсчетом текущего суммарного значения числа aw элементов $C_{ij} = 1$, расположенных выше диагонали, что позволяет определить элемент $WJI[aw] = j$. Формирование координатной матрицы ERC , содержащей относительные адреса входа в строку, осуществляется в одном цикле с формированием WJI путем нахождения относительного адреса aw , при этом сканирование i -й строки определяет значение $ERC[i] = aw + 1$.

Одновременно с этими массивами функция $cord$ должна формировать массив SI , содержание которого отображает структуру нулевого столбца исходной матрицы. Относительный адрес элемента этого массива as подсчитыва-

ется путем вычисления числа значащих элементов в этом столбце. Функция *cord* также заполняет массив *P* содержанием нулевой строки индексной матрицы *C*. Это необходимо делать потому, что после формирования координатного описания матрица *C* удаляется из памяти либо на очередном витке уборки мусора, либо в явной форме путем вызова соответствующей системной функции.

Модификация программного обеспечения на этапе численного анализа

Перед началом формирования численного описания компонентов моделируемой схемы следует открыть массивы *WD*, *WU*, *WL* для хранения компактного описания, при этом все указанные массивы должны быть обнулены до их использования. Каждый тип компонента моделируемой электронной схемы требует разработки своего уникального алгоритма, однако все эти алгоритмы могут быть построены в соответствии с единообразной методикой, которая рассматривается далее.

Исходными данными для формирования описания компонентов типа *спр* являются массивы включения компонентов *in_спр* и значений параметров *z_спр*, кроме того, известно содержание массивов *WJI* и *ERC*. Конечной целью является описание компонентов в компактных массивах *WD*, *WU*, *WL*, для чего необходимо решить две вспомогательные задачи.

Во-первых, необходимо определить новые значения *ii* индексов строк и столбцов *jj*, которые получаются после многократных перестановок этих строк и столбцов в процессе их оптимального упорядочивания. Такая задача решается путем организации цикла, в котором реализуется перекодировка индексов всех переменных согласно содержанию вектора перестановок *P*.

Второй задачей, которая возникает на этапе численного формирования, является определение относительного адреса, т. е. местоположения элемента в массивах *WL*, *WU*, куда необходимо занести значение параметра w_{ij} описываемого компонента. Так, если требуется найти относительный адрес для некоторого элемента w_{ij} , для которого значения индексов строк и столбцов изменены с *i* на *ii* и с *j* на значение *jj*, то такая задача может быть решена путем включения программного обеспечения цикла, по выходе из которого значение относительного адреса *a* будет соответствовать эле-

менту, расположенному в (*ii*)-й строке и (*jj*)-м столбце. Вследствие структурной симметрии, этот же относительный адрес определяет положение элемента в *ii*-м столбце и *jj*-й строке ниже диагонали.

В соответствии с изложенной стратегией нетрудно построить совокупность функций формирования описания компонентов для всех компонентов *спр* схемы. Структура таких функций в целом будет повторять структуру, аналогичную соответствующим функциям формирования полной матрицы и отличается от последних только наличием приведенных выше фрагментов для выполнения перекодировки и нахождения относительных адресов.

При решении уравнений моделируемой схемы на численном этапе компактной обработки в качестве основы для численной *LU*-факторизации можно принять блок-схему (рис. 2), которая позволяет наиболее просто осуществить расчет элементов матриц *L* и *U* по виду массива полной матрицы *W* схемы. В результате реализации приведенной блок-схемы в массиве *w* отображается как элемент матрицы *L*, так и элементы матрицы *U*, при этом в поддиагональной части и на диагонали будут содержаться элементы матрицы *L*, а в наддиагональной части — элементы матрицы *U-E* (где *E* — единичная матрица).

Однако, поскольку массив *w* при компактной обработке отсутствует, то необходимо разработать алгоритм поиска элементов w_{kk} , w_{ki} , w_{kj} , w_{ij} непосредственно в массивах *WD*, *WU*, *WL*.

Элемент w_{kk} легко выделяется из компактного массива *WD* по известному значению переменной *k*, т. е. $w_{kk} = WD[k]$.

Элементы w_{kj} и w_{ik} находятся сканированием массива *WJI* в границах $a_{k1} = ERC[k]$,

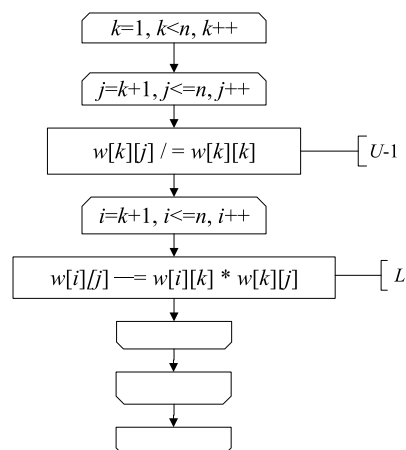


Рис. 2. Блок-схема численной *LU*-факторизации

$a_{k2} = ERC[k + 1] - 1$, что позволяет найти относительные адреса a_i и a_j расположения индексов i и j .

Таким образом, имеем соотношения для поиска элемента w_{kj} :

$$j = WJI[a_j], w_{ij} = WU[a_j],$$

и аналогичные соотношения для поиска элемента w_{ik} :

$$i = WJI[a_i], w_{ik} = WL[a_i].$$

В отличие от элементов w_{kj} и w_{ik} , расположение которых относительно диагонали заранее известно, элемент w_{ij} может находиться в любом месте j -го столбца. При этом возможны три случая

1) $i = j$, при этом элемент следует искать в массиве WD , т.е. $w_{ii} = WD[i]$;

2) $i < j$, при этом поиск надо осуществлять в i -й строке j -го столбца, начиная с относительного адреса $a_i = ERC[i]$, и искомым относительный адрес находится в пределах границ a_{i1} и a_{i2} , элемент извлекается из массива WU ;

3) $i > j$, поиск необходимо осуществлять в j -м столбце и i -й строке. Так как ниже диагонали элементы расположены по столбцам, то относительный адрес входа в столбец определяется выражением $a_j = ERC[j]$. Поскольку искомым относительный адрес находится в пределах границ a_{i1} и a_{i2} , то зная значение относительного адреса, можно извлечь искомым элемент из WL .

Таким образом, численная LU -факторизация может быть осуществлена на основе компактных численных массивов WD , WU , WL и координатных массивов ERC , WJI без отображения в памяти компьютера полной разреженной матрицы.

В результате LU -факторизации исходное матричное уравнение электронной схемы

$$WX + S = 0$$

распадается на два уравнения

$$\begin{aligned} UX &= Q, \\ LQ + S &= 0, \end{aligned}$$

где X — вектор переменных исходного уравнения; S — вектор задающих источников; U — верхняя треугольная матрица; L — нижняя треугольная матрица; Q — вспомогательный вектор переменных.

При расчете переменных сначала выполняется прямой ход, в процессе которого решается второе уравнение относительно вспомогательного вектора Q , затем решается первое уравнение, что дает искомые значения вектора X .

В процессе прямого хода выполняется последовательный расчет переменных, образующих вспомогательный вектор Q :

$$q_1 = s_1/l_{11}, q_i = -\left(s_i + \sum_{j=1}^{j=i-1} l_{ij}q_j\right)/l_{ii}, i = \overline{(2, n)}.$$

Приведенная форма реализации прямого хода является общепринятой при полном описании моделируемых систем разреженными матрицами, однако она является крайне неудобной при использовании ее в случае компактного описания на основе строчно-столбцового фиксированного формата. Это объясняется тем, что приведенные соотношения требуют выполнения сканирования матрицы L по строкам. Вместе с тем в строчно-фиксированном формате элементы, расположенные ниже диагонали, формируются в компактных матрицах по столбцам, и, следовательно, сканирование компактной матрицы WL , отображающей в компактной форме нижнюю треугольную матрицу L , оказывается невозможным. Следовательно, возникает задача разработки нового алгоритма реализации прямого хода, который будет осуществлять сканирование не по строкам, а по столбцам.

Для модификации алгоритма прямого хода запишем уравнение прямого хода $LQ + S = 0$ в виде

$$DQ + (L - D)Q + S = 0, \quad (1)$$

где D — диагональная матрица, образованная из диагональных элементов матрицы L .

Введем вспомогательный вектор T , определяемый выражением

$$T = (L - D)Q.$$

Тогда уравнение (1) может быть записано в виде

$$DQ + T + S = 0. \quad (2)$$

Решение уравнения (2) имеет вид

$$Q = D^{-1}(T + S). \quad (3)$$

Составляющие вектора T можно формировать путем накопления элементов по мере вы-

числения очередных значений. При этом i -я составляющая t_i вектора получает приращение $\Delta t_i = l_{ij}q_j$ ($i = \overline{j+1, n}$).

Естественно, перед началом указанной процедуры начальные значения вектора T должны быть обнулены. Используя обновленное на очередном шаге значение вектора T , можно последовательно рассчитывать значения составляющих вектора Q , используя скалярное представление уравнения (3):

$$q_j = -(t_j + s_j)/l_{jj}, \quad (j = \overline{1, n}).$$

При этом для $j = 1$ имеем $t_j = 0$ и $q_1 = -s_1/l_{11}$.

Далее, найдем $t_2 = l_{21}q_1$, что позволяет вычислить следующую составляющую вектора вспомогательных переменных:

$$q_2 = -(t_2 + s_2)/l_{22}.$$

Аналогичным образом вычисляются все остальные составляющие вектора вспомогательных переменных Q , который используется в качестве задающего вектора при выполнении обратного хода.

Естественно, что изложенный алгоритм в силу своей относительной сложности нецелесообразно рекомендовать при полном описании схемы разреженными матрицами. Однако этот алгоритм выполнения прямого хода LU -факторизации, в отличие от общеизвестного алгоритма, реализует сканирование исходной матрицы по столбцам. Именно такая организация сканирования является единственно возможным способом поиска элементов поддиагональной части исходной матрицы при ее компактном представлении в строчно-столбцовом фиксированном формате.

На обратном ходе осуществляется решение уравнения $UX = Q$, при этом для организации вычислительного процесса необходимо осуществлять сканирование по строкам. Поскольку в строчно-столбцовом фиксированном формате наддиагональные элементы записываются по строкам, то нет необходимости при переходе к компактной обработке менять общепринятую стратегию построения алгоритма реализации обратного хода. Необходимо лишь организовать сканирование i -й строки в границах $a_1 = ERC[i]$, $a_2 = ERC[i+1] - 1$.

Поскольку в процессе LU -факторизации на этапе индексного анализа выполнялось оптимальное упорядочивание строк и столбцов в соответствии с алгоритмом минимальной

степени, фактическая нумерация последовательности строк и столбцов отличается от исходной. Следовательно, при окончательном расчете переменных моделируемой системы необходимо осуществить перекодировку вектора X согласно содержанию вектора перестановок P .

Заключение

Рассмотренная в статье индексно-численная технология сжатия данных основана на использовании двухэтапной процедуры построения компактного описания, включающей в себя индексный и численный этапы. При этом на индексном этапе уточняется формат описания схемы с учетом возможного появления новых ненулевых элементов в процессе решения уравнений схемы, а на численном этапе в уточненном формате формируется компактное описание моделируемой схемы без предварительного построения ее полных разреженных матриц. На заключительном этапе выполняется решение уравнений методом LU -факторизации, при этом расчет переменных осуществляется на основе компактных численных массивов WD , WU , WL и координатных массивов ERC , WJI без отображения в памяти компьютера полной разреженной матрицы.

Практическая реализация двухэтапного процесса сжатия данных позволяет существенно повысить эффективность работы систем автоматизированного проектирования.

Список литературы

1. Анисимов В. И., Гридин В. Н. Методы построения систем автоматизированного проектирования на основе Интернет-технологий и компактной обработки разреженных матриц // Информационные технологии в проектировании и производстве. 2009. № 1. С. 3—7.
2. Коваленко О. С., Курейчик В. М. Обзор проблем и состояний облачных вычислений и сервисов // Известия ЮФУ. Технические науки. 2012. № 7. С. 146—153.
3. Гридин В. Н., Дмитриевич Г. Д., Анисимов Д. А. Построение систем автоматизированного проектирования на основе web-технологий // Информационные технологии. 2011. № 5. С. 23—27.
4. Анисимов Д. А. Методы построения систем автоматизации схмотехнического проектирования на основе веб-сервисов // Известия СПбГЭТУ "ЛЭТИ". 2012. № 10. С. 56—61.
5. Писсанецки С. Технология разреженных матриц. М.: Мир, 1988. 406 с.
6. Эстербю О., Златев З. Прямые методы для разреженных матриц. М.: Мир, 1987.
7. Кнут Д. Искусство программирования для ЭВМ. Т. 1. М.: Мир, 1976, 734 с.

8. Баталов Б. В., Егоров Ю. Б., Русаков С. Г. Основы математического моделирования больших интегральных схем на ЭВМ Основы математического моделирования больших интегральных схем на ЭВМ. М.: Радио и связь. 1982. 168 с.
9. Джордж А., Лю Д. Численное решение больших разреженных систем уравнений. М.: Мир, 1984. 333 с.
10. Гридин В. Н., Анисимов В. И., Абухазим М. М. Сжатие данных в системах автоматизации схемотехнического

проектирования на основе методов фиксированного формата // Системы высокой доступности. 2016. № 4. С. 34–40.

11. Влах И., Сингхал К. Машинные методы анализа и проектирования электронных схем. М.: Радио и связь, 1988. 560 с.

12. Чуа Л. О., Лин Пен-Мин. Машинный анализ электронных схем. М.: Энергия, 1980. 631 с.

13. Норенков И. П. Введение в автоматизированное проектирование технических устройств и систем. М.: Высшая школа, 1986.

V. N. Gridin¹, Scientific Director, D. Sc., Professor, e-mail: info@ditc.ras.ru,
V. I. Anisimov^{1,2}, Chief Researcher, D. Sc., Professor, e-mail: vianisimov@inbox.ru,
M. M. Abuhazim², Postgraduate student, e-mail: Abuhazim_monther@Yahoo.com,

¹ Design information technologies Center Russian Academy of Sciences, Odintsovo, Russian Federation,

² Saint-Petersburg Electrotechnical University, Saint-Petersburg, Russian Federation

Index-Numerical Technology of Data Compression in Automation Systems of Circuit Design

The ways to increase the productivity of computer-aided design based on the transition to a compact form of storage and processing of sparse matrices. The description of the most effective method of compact storage and processing of sparse matrices on the basis of a fixed row-column format is given. It is shown that a distinctive feature of the fixed format methods is the impossibility of including additional non-zero elements in the description, which excludes the possibility of direct application of these methods for information processing in solving systems of equations by any numerical method, due to the inevitable appearance of new non-zero elements in the process of this solution. To be able to take into account new non-zero elements in the compact description of the simulated system, it is proposed to use the index-numerical technology for the formation of such a description and a description of the implementation of data compression based on the division of the General procedure into two independent parts of the index and numerical analysis. The technique of construction of the software of systems of automation of circuit design on the basis of index-numerical technology of data compression is given.

Keyword: computer-aided design systems, system modeling, compact processing, sparse matrices, row-column fixed format

DOI: 10.17587/it.25.349-357

Reference

1. Anisimov V. I., Gridin V. N. *Informacionnye tekhnologii v proektirovanii i Proizvodstve*, 2009, no. 1, pp. 3–7 (in Russian).
2. Kovalenko O. S., Kurejchik V. M. *Izvestiya YUFU. Tekhnicheskie Nauki*, 2012, no. 7, pp. 146–153 (in Russian).
3. Gridin V. N., Dmitrevich G. D., Anisimov D. A. *Informacionnye Tekhnologii*, 2011, no. 5, pp. 23–27 (in Russian).
4. Anisimov D. A. *Izvestiya SPbGEHTU "LEHTI"*, 2012, no. 10, pp. 56–61 (in Russian).
5. Pissanecki C. *Sparse Matrix Technology*, Moscow, Mir, 1988, 406 p. (in Russian).
6. Ehsterbyu O., Zlatev Z. *Direct methods for sparse matrices*, Moscow, Mir, 1987 (in Russian).
7. Knut D. *The art of computer programming*, vol. 1, Moscow, Mir, 1976. 734 p. (in Russian).

8. Batalov B. V., Egorov Yu. B., Rusakov S. G. *Fundamentals of mathematical modeling of large integrated circuits on a computer*, Moscow, Radio i svyaz', 1982, 168 p. (in Russian).

9. Dzhordzh A., Lyu D. *Numerical solution of large sparse systems of equations*, Moscow, Mir, 1984, 333 p. (in Russian).

10. Gridin V. N., Anisimov V. I., Abuhazim M. M. *Sistemy vysokoj dostupnosti*, 2016, no. 4, pp. 34–40 (in Russian).

11. Vlah I., Singhal K. *Machine methods of analysis and design of electronic circuits*, Moscow, Radio i svyaz', 1988, 560 p. (in Russian).

12. Chua L. O., Lin Pen-Min. *Machine analysis of electronic circuits*, Moscow, EHnergiya, 1980, 631 p. (in Russian).

13. Norenkov I. P. *Introduction to computer-aided design of technical devices and systems*, Moscow, Vysshaya shkola, 1986 (in Russian).