

УСКОРЕНИЕ ВЫЧИСЛЕНИЙ В ЗАДАЧАХ РАСЧЕТА ЗАЗЕМЛИТЕЛЕЙ

Д.С. Шишигин

*Научный руководитель С.Л. Шишигин, д-р техн. наук, доцент
Вологодский государственный университет
г. Вологда*

Заземляющее устройство (ЗУ), состоящее из заземлителя и всех подключенных к нему проводников в воздухе, предназначено для растекания токов молнии, короткого замыкания, импульсных помех в землю. Расчет ЗУ – трудоемкая задача с численным решением. Быстродействие компьютерной программы – один из основных критериев, определяющих ее конкурентоспособность, косвенный признак эффективности заложенных в неё алгоритмов. В данной работе рассматриваются методы и средства повышения производительности вычислений в программе ZYM.

При численных расчетах ЗУ дробится на элементы (короткие стержни), число которых составляет тысячи, десятки тысяч для современных подстанций. Все элементы связаны между собой гальваническими (через землю) и электромагнитными связями. Каждый элемент моделируется П–четырёхполюсником с RLC элементами, поэтому математической моделью ЗУ является цепная схема замещения, а параметры ЗУ представлены квадратными, полностью заполненными матрицами сопротивлений растеканию \mathbf{R} , индуктивностей \mathbf{M} , и емкостей \mathbf{C} [1]. Расчет напряжений и токов в этой схеме производится методами теории электрических цепей. При найденных токах электромагнитное поле ЗУ рассчитывается методами теории антенн. Проблема в том, что эти расчеты, включающие операции умножения, обращения действительных и комплексных матриц, очень трудоемки. Таких операций может быть много. Действия с комплексными числами увеличивают время в 4 раза, поэтому существующие программы расчета ЗУ вынуждены ограничивать размерность задач, либо проводить расчеты без учета взаимных параметров.

Алгоритмическая оптимизация кода – один из основных способов повышения производительности. Согласно [2] 50% времени счета связаны с 4% кода, поэтому именно эти «узкие» места следует найти для оптимизации. В решаемой задаче это расчет коэффициентов матрицы и операции с матрицами. При $N=10000$ требуется вычислить $3 \cdot 10^8$ коэффициентов массивов $\mathbf{R}, \mathbf{M}, \mathbf{C}$, поэтому, следуя рекомендациям [2], можно повысить быстродействие.

Идеи по оптимизации кода матричных операций находим в тексте программ математического пакета *Alglib* (распространяется свободно). Например, если вместо привычного (по учебникам математики) алгоритма умножения матриц

(рис.1а) применить алгоритм (рис.1б), где вычисления индексов двумерных матриц заменены действиями с указателями, то быстроедействие возрастает в 7 раз (зависит от компьютера, языка программирования и т.д.), что существенно.

```
//A[N,L]*B[L,M]=C[N,M]
for i:=0 to N do
  for j:=0 to M do
    begin
      v:=0;
      for k:=0 to L do
        v:=v+A[i,k]*B[k,j];
      C[i,j]:=v;
    end;
```

Рис.1а. Стандартный код программы перемножения матриц

```
for i:=0 to N do
  for j:=0 to L do
    begin
      pC:=@C[i][0]; pB:=@B[j][0]; v:=A[i,j];
      for k:=0 to M do
        begin
          pC^:=pC^+v*pB^; Inc(pC); Inc(pB);
        end;
    end;
```

Рис.1б. Оптимизированный код программы перемножения матриц (пакет Alglib)

Заметим, однако, что алгоритмическая оптимизация обычно идет в ущерб простоте и читабельности кода и может привести к отрицательным результатам. В [2] отмечено, что применение оптимизирующего компилятора для простого кода может дать бóльший эффект, чем «хитрая» оптимизация циклов, после которой компилятор бессилён ускорить вычисления.

Для решения задач линейной алгебры используются специализированные математические пакеты. К их числу относится коммерческая библиотека *Intel Math Kernel Library* (далее *Intel MKL*), которая предназначена для математических расчетов с высокой производительностью. В состав *Intel MKL* входят программы для работы с матрицами, быстрое преобразование Фурье, векторная математическая и статистические библиотеки, функции расширенной точности, программы для решения дифференциальных уравнений и методы оптимизации. Пользователи популярных математических пакетов, таких как *Mathcad* и *Matlab* вряд ли знают, что высокому быстроедействию матричных операций они обязаны *Intel MKL*.

Данный пакет может быть подключен к собственной программе как любая динамическая библиотека. Рекомендуемые языки программирования – FORTRAN, на котором и создан пакет *Intel MKL*, а также C, C++, для которых разработаны интерфейсы. Эту библиотеку можно подключить также из программы, написанной на других языках, однако это потребует от программиста дополнительных усилий.

Библиотека *Intel MKL* является потоко-безопасной, поддерживает распараллеливание и оптимизирована под многоядерные системы. Используя *Intel MKL*, разработчик может повысить производительность своего приложения за счет многопоточности и низкоуровневой оптимизации алгоритмов. Функции библиотеки оптимизированы для работы на процессорах *Intel*, однако

они конкурентоспособны и на процессорах других производителей, что делает их универсальными. Существует 32- и 64-разрядная версия *Intel MKL*.

Программирование с использованием *Intel MKL* имеет специфику. Мы уже отмечали снижение производительности из-за индексирования элементов больших двумерных массивов (рис.1). В *Intel MKL* используются только одномерные массивы, имеющие непрерывное расположение в оперативной памяти. Поэтому привычные двумерные массивы перед обращением к функциям *Intel MKL* нужно преобразовать в одномерные массивы, что требует дополнительной памяти, либо сразу записывать матрицы в одномерные массивы. Следует помнить, что, следуя правилам языка Фортран, двумерные массивы заполняются по столбцам.

Сопоставим быстродействие *Intel MKL* с аналогичным, но свободно распространяемым пакетом *Alglib* для основных матричных операций (табл.). Размерность матриц – 2000. Расчеты проведены на ноутбуке: Windows 7 64-bit, Intel Core i7 4x2.2 ГГц, ОЗУ 6 Гб.

Таким образом, применение *Intel MKL* существенно повышает быстродействие матричных вычислений по сравнению с пакетом *Alglib* и аналогичных. При разработке коммерческих программ с матричными операциями он, несомненно, стоит затраченных средств. Дополнительным доводом к его применению являются «рекомендации» Mathcad и Matlab. Единственное сожаление вызывает отсутствие в составе библиотеки итерационных методов линейной алгебры.

Таблица

Сравнение пакетов *Alglib* и *Intel MKL*

Операция	AlgLib	Intel MKL	Сравнение
Умножение матриц $[A] \cdot [A]$	11.8 сек	0.6 сек	19 раз
Решение системы $[A] \cdot [X] = [B]$	3.9 сек	0.3 сек	13 раз
Обращение матрицы $[A]^{-1}$	28.8 сек	1.1 сек	26 раз
Решение системы $[C] \cdot [X] = [B]$ с комплексными матрицами	46 сек	0.9 сек	51 раз
Обращение комплексной матрицы $[C]^{-1}$	125 сек	3.6 сек	34 раз

В качестве примера рассчитаем сопротивление заземлителя (квадратная сетка со стороной 100 м) при уменьшении размера ячейки, что приводит к увеличению числа элементарных стержней N . Нам необходимо рассчитать коэффициенты матрицы собственных и взаимных коэффициентов R , число которых равно N^2 , а затем решить СЛАУ для нахождения токов элементов и сопротивления заземлителя. С увеличением N время счета возрастает (рис.2), но существенно медленнее, чем квадратная парабола при расчете коэффициентов и кубическая пара-

бола при решении СЛАУ, что объясняется многопоточностью вычислений и рассмотренными характеристиками *Intel MKL*.

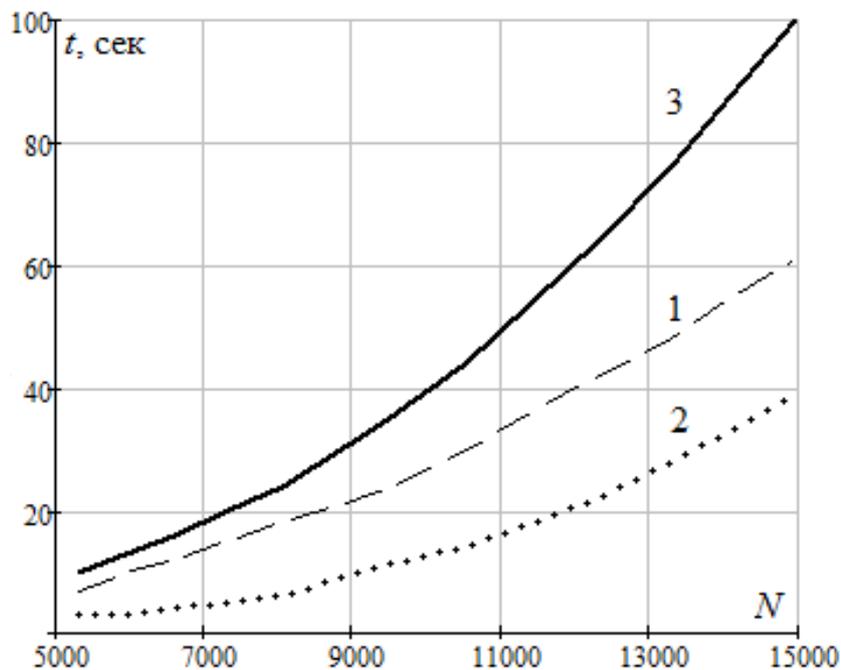


Рис. 2. Время расчета коэффициентов матрицы размерности N (1), решения системы уравнений (2), суммарное (3)

Вывод. Быстродействие компьютерной программы – один из основных критериев, определяющих ее конкурентоспособность, косвенный признак эффективности заложенных в ней алгоритмов. Существует резервы повышения быстродействия программы за счет оптимизации кода, распараллеливания алгоритмов, многопоточности вычислений. В задачах с матричными операциями большой размерности эффективным средством повышения производительности расчетов является применение математической библиотеки *Intel Math Kernel Library*.

ЛИТЕРАТУРА

1. Шишигин, С.Л. Расчет заземлителей: учебное пособие / С.Л. Шишигин. – Вологда: ВоГТУ, 2012. – 121 с.
2. Макконнелл, С. Совершенный код : практическое руководство по разработке программного обеспечения / С. Макконнелл; [пер. с англ.]. - М.: Русская редакция, 2014. - 896 с.