Министерство образования и науки Российской Федерации

Новосибирский государственный технический университет

Кафедра прикладной математики

Программирование вычислений

Лабораторная работа №3

Факультет: ПМИ

Группа: ПМИ-31

Студенты: Челдаев А.В.

Елфимов М.С.

Мазин С.С.

Преподаватели: Рояк М.Э.

Рояк С.Е.

Вариант: 9.1

Новосибирск

2015

1. Цели работы

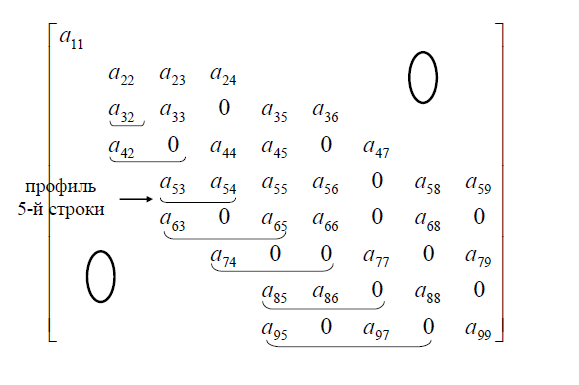
Изучение псевдодинамического распределения памяти на языке ФОРТРАН; изучение форматов хранения матриц большой размерности; оптимизация программ по точности, скорости, памяти; изучение погрешности вычисления скалярного произведения; изучение способов отладки; изучения принципов формирования тестов для вычислительных программ; изучение файлов прямого доступа.

1. Задание

Написать программу, реализующую на языке ФОРТРАН требуемые действия над матрицами произвольной (задаваемой пользователем) размерности.

Вариант 9.1. Умножение симметричной матрицы в профильном формате на вектор. Хранится только нижний треугольник по строкам. С выделением диагонали.

1. Анализ задачи

Профильный формат хранения матриц в некотором смысле является обобщением ленточного формата на случай, когда матрица не обладает строго диагональной структурой и ненулевые элементы расположены в произвольном порядке, но при этом они сосредоточены у главной диагонали (т.е. в каждой строке матрицы ширина ленты будто бы разная).

В такой ситуации в каждой строке (столбце) можно выделить профиль строки (столбца) – это количество элементов строки (столбца) от первого ненулевого элемента в строке (столбце) до диагонального элемента (не включая его).

Реализована идея псевдодинамической памяти, т.е. выделяется один массив, а все массивы, необходимые для вычислений, размещены в нем; доступ к ним осуществляется путем доступа с необходимым смещением.

Умножение матрицы реализуется следующим образом: диагональные элементы матрицы умножаются на соответствующие элементы вектора, далее профиль строки умножается на соответствующие элементы вектора, полученные результаты прибавляются к соответствующим элементам результирующего вектора. После завершения умножения полученный результирующий вектор будет являться результатом умножения матрицы на вектор.

Входные данные:

AL.txt – элементы матрицы нижнего треугольника, хранящиеся по строкам.

DI.txt – диагональные элементы матрицы.

IA.txt – информация о профиле.

Nmatrix.txt – размер матрицы.

Nvector.txt – размер вектора.

Vector.txt – вектор.

Матрица может быть выведена в плотном формате в текстовый файл.

1. Текст программы

program laba3

common/vecsize/ivs

common/size/isMat,dal,iFsize

dimension A(380616684)

iFsize=380616684

open (10, file = 'AL.txt', err=1)

open (11, file = 'DI.txt', err=1)

open (12, file = 'IA.txt', err=1)

open (14, file = 'Nmatrix.txt', err=1)

open (15, file = 'Nvector.txt', err=1)

open (16, file = 'Vector.txt', err=1)

call Input(A)

call Mult(A)

goto 2

1 pause 'Error opening file!'

stop

2 close(10)

close(11)

close(12)

close(13)

close(14)

close(15)

close(16)

print\*, 'All done :)!'

pause

end

**\*\*Умножение матрицы на вектор\*\***

subroutine Mult(A)

common/size/isMat,dal,iFsize

common/vecsize/ivs

dimension A(\*)

idlvec=2\*isMat+2+dal -1

idldi=isMat+2 -1

idlal=2\*isMat+2 -1

idlrez=2\*isMat+1+dal+ivs

do i=1,isMat

A(idlrez+i)=A(idldi+i)\*A(idlvec+i)

enddo

do i=1,isMat

prof=A(i+1)-A(i)

do j=A(i),A(i+1)-1

ic = idlvec+i-prof

A(idlrez+i)=A(idlrez+i)+A(idlal+j)\*A(ic)

A(ic-idlvec+idlrez)=A(ic-idlvec+idlrez)+A(idlal+j)\*A(idlvec+i)

prof=prof-1

enddo

enddo

call Output(A)

end

**\*\*Проверка на корректность данных\*\***

subroutine Check(X)

common/size/isMat,dal,iFsize

common/vecsize/ivs

if (ivs.lt.1.or.ivs.ne.isMat)then

pause 'Invalid input data!'

stop

endif

if ((2\*isMat+3\*ivs+1+X).gt.iFsize)then

pause 'Not enough memory!'

stop

endif

goto 2

1 pause 'The problem with reading from a file!'

stop

2 end

**\*\*Вывод матрицы в файл\*\***

subroutine Output(A)

common/size/isMat,dal,iFsize

common/vecsize/ivs

dimension A(\*)

10 format (F9.2\)

open (1, file = 'REZULTAT.txt', err=20)

write(1,10) (A(i),i = 2\*isMat+1+dal+ivs+1,2\*isMat+1+dal+ivs

\*+isMat ,1)

goto 30

20 pause 'Error opening file!'

stop

30 end

**\*\*Считывание данных\*\***

subroutine Input(A)

common/size/isMat,dal,iFsize

common/vecsize/ivs

dimension A(\*)

10 format (F5.2)

read (14,\*) isMat

if (isMat.lt.iFsize.and.isMat.gt.1)then

read (15,\*) ivs

read (12,\*,end=5) (A(i),i = 1, isMat+1,1) !IA

5 dal = A(isMat+1)-1

call Check(dal)

read (11,\*,end=6) (A(i),i = isMat+2, 2\*isMat+1,1) !DI

6 read (10,\*,end=7) (A(i),i = 2\*isMat+2, 2\*isMat+1+dal,1) !AL

7 read (16,\*,end=8) (A(i),i = 2\*isMat+2+dal , 2\*isMat+1+dal+ivs,1) !VEC

else

pause 'Not enough memory or size of matrix < 1 !'

stop

endif

8 end

Текст программы для вывода матрицы в плотном формате

program conv\_matrix

dimension A(1000)

common/size/isMat,dal

common/vecsize/ivs

open (10, file = 'AL.txt', err=1)

open (11, file = 'DI.txt', err=1)

open (12, file = 'IA.txt', err=1)

open (13, file = 'Nmatrix.txt', err=1)

call Input(A)

call Convert(A)

goto 2

1 pause 'Error opening file!'

stop

2 close(10)

close(11)

close(12)

close(13)

pause 'All done :) !'

end

**\*\*Преобразование матрицы\*\***

subroutine Convert(A)

common/size/isMat,dal

common/vecsize/ivs

dimension A(\*)

1 format(F5.2\)

2 format()

dimension rez(1000)

idldi=isMat+2 -1

idlal=2\*isMat+2 -1

imove=0

do i=1,isMat\*isMat

rez(imove+i)=A(idldi+i)

imove=imove+isMat

enddo

imove=0

do i=1, isMat

iprof=A(i+1)-A(i)

ic=isMat\*(i-iprof-1)+i

do j=A(i),A(i+1)-1

rez(imove+i-iprof)=A(idlal+j)

rez(ic)=A(idlal+j)

ic=ic+isMat

iprof=iprof-1

enddo

imove=imove+isMat

enddo

call Output(rez)

end

**\*\*Вывод данных\*\***

subroutine Output(rez)

common/size/isMat,dal

common/vecsize/ivs

dimension rez(\*)

10 format (F9.2\)

2 format()

open (1, file = 'REZULTAT.txt', err=20)

do i=1,isMat\*isMat

write(1,10) rez(i)

if (mod(i,isMat).eq.0.0)then

write(1,2)

endif

enddo

goto 30

20 pause 'Error opening file!'

stop

30 end

**\*\*Считывание данных\*\***

subroutine Input(A)

common/size/isMat,dal

common/vecsize/ivs

dimension A(\*)

10 format (F5.2)

read (13,\*) isMat

read (12,\*,end=5) (A(i),i = 1, isMat+1,1) !IA

dal = A(isMat+1)-1

5 read (11,\*,end=6) (A(i),i = isMat+2, 2\*isMat+1,1) !DI

6 read (10,\*,end=7) (A(i),i = 2\*isMat+2, 2\*isMat+1+dal,1) !AL

7 end

1. Тесты

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Результат | Назначение |
| 1 | al = {7 0 5 1 4}  di = {6 7 5 4 3}  ia = {1 1 1 3 5 6}  Nmatrix = 5  Nvector = 5  Vector = {5 12 7 8 4} | 79.00 124.00 78.00 115.00 44.00 | Общая проверка работоспособности программы |
| 2 | al = {7 0 5 1 4}  di = {6 7 5 4 3}  ia = {1 1 1 3 5 6}  Nmatrix = 6  Nvector = 5  Vector = {5 12 7 8 4} | Invalid input data! | Размеры матрицы и вектора не совпадают |
| 3 | al = {7 0 5 1 4}  di = {6 7 5 4 3}  ia = {1 1 1 3 5 6}  Nmatrix = 93500000  Nvector = 93500000  Vector = {5 12 7 8 4} | Not enough memory! | Нехватка памяти |