|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Прикладной математики | | |
|  | | |
| Лабораторная работа № 3 | | |
| по дисциплине «Программирование вычислений» | | |
|  | | |
| **Алгоритмы и структуры данных****для работы с разреженными матрицами** | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМ-81 |
| Вариант: | 12.1 |
| Студенты: | Басов Денис, |
|  | Юрганов Егор |
| Бригада: | 4 |
| Преподаватель: | Рояк Михаил Эммануилович. |
|
|  |  |
| Новосибирск | | |
| 2020 | | |

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; изучение принципов формирования тестов для вычислительных программ; изучение файлов прямого доступа.

1. **Условие задачи**

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

⦁ Реализовать задание лабораторной работы с использованием написанных ранее подпрограмм при условии, что матрицы и векторы хранятся в файлах прямого доступа (файл с размерностями оставить текстовым).

⦁ Разработать программу генерации тестов большой размерности в заданном формате. Программа должна создавать все требуемые файлы прямого доступа.

⦁ Для просмотра и создания тестов с файлами прямого доступа рекомендуется написать дополнительные программы, которые формируют файл прямого доступа по заданному текстовому файлу, и наоборот.

Вариант 12.1. Умножение симметричной матрицы на вектор. Матрица в памяти хранится в разреженном формате с выделенной диагональю (строчно-столбцовый формат).

1. **Анализ задачи**

Разреженный формат хранения матриц используется, когда в матрице ненулевые элементы расположены в произвольном порядке. Хранятся не все элементы, а только ненулевые элементы. Поэтому дополнительно используется еще один массив, содержащий информацию о втором индексе внедиагонального элемента.

В данной задаче нам дана матрица с симметричным относительно главной диагонали портретом, то есть местоположение ненулевых элементов симметрично.

Строчно-столбцовый формат подразумевает хранение элементов нижнего треугольника по строчкам, верхнего по столбцам. Исходя из условий задачи можно хранить только один треугольник(нижний), так как нижний и верхний треугольники совпадают.

Умножение матриц реализуется следующим способом: умножается каждая строка на соответствующий элемент вектора и этот результат прибавляется к значению диагонали до тех пор, пока строка не закончится.

1. **Текст программы**

Программа умножения матрицы на вектор

program main

implicit none

integer\*4 n,k,readia

real a

dimension a(50000000)

open(2,file='size.txt')

read(2,\*)n

k = readia(a(1),n)

close(2)

call readother(a(n+2),a(n+n+2),a(2\*n+2+k),

,a(2\*(n+k+1)),n,k)

call mult(a(1), a(n+2), a(n+n+2), a(2\*n+2+k),

,a(2\*(n+k+1)),a(2\*(n+k+1)+n), n, k)

end

function readia(ia,n)

integer i,ia

dimension ia(n+1)

open(3, file = 'ia.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

do i=1,n + 1

read(3, rec = i, err = 100) ia(i)

enddo

close(3)

readiad = int(ia(n+1)) - 1

return

100 pause 'error'

end

subroutine readother(di,ja,al,v,n,k)

integer i,k,n,ja

real di,al,v

dimension di(n),ja(k),al(k),v(n)

open(4, file = 'ja.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

do i=1,k

read(4, rec = i, err = 100) ja(i)

end do

close(4)

open(4, file = 'di.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

do i=1,n

read(4, rec = i, err = 100) di(i)

end do

close(4)

open(4, file = 'al.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

do i=1,k

read(4, rec = i, err = 100) al(i)

end do

close(4)

open(4, file = 'vector.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

do i=1,n

read(4, rec = i, err = 100) v(i)

end do

close(4)

return

100 pause 'error'

end

subroutine mult(ia,di,ja,al,x,y,n,c)

integer i,k,c,ia,ja

real di,al,x,y

dimension ia(n+1),ja(c),di(n),al(c),x(n),y(n)

do i=1,n

y(i) = di(i)\*x(i)

enddo

do i=1,n

do k=ia(i),ia(i+1)-1

y(i) = y(i) + al(k)\*x(ja(k))

y(ja(k)) = y(ja(k)) + al(k)\*x(i)

end do

enddo

open(5,file='result.txt')

write(5,\*)(y(i),' ',i=1,n)

close(5)

return

end

Программа перевода bin в txt

program main

implicit none

integer n, k

open(10, file = 'size.txt', err = 100)

read(10, \*, err = 100) n

close(10)

call bin2txtint('ia.txt', 'ia.bin', n + 1)

open(10, file = 'ia.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

read(10, rec = n + 1, err = 100) k

call bin2txtint('ja.txt', 'ja.bin', k-1)

call bin2txtreal('di.txt', 'di.bin', n)

call bin2txtreal('al.txt' , 'al.bin', k-1)

call bin2txtreal('vector.txt' , 'vector.bin', n)

return

100 pause 'Error'

end

subroutine bin2txtreal(txt, bin, n)

integer i, n

character \*(\*) txt, bin

real a

dimension a(100000)

open(11, file = txt, err = 100)

open(12, file = bin, form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

do i = 1, n

read(12, rec = i, err = 100) a(i)

end do

write(11, \*, err = 100) (a(i), i = 1, n)

close(11)

close(12)

return

100 pause 'Error'

end

subroutine bin2txtint(txt, bin, n)

integer i, n

character \*(\*) txt, bin

integer a

dimension a(100000)

open(11, file = txt, err = 100)

open(12, file = bin, form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

do i = 1, n

read(12, rec = i, err = 100) a(i)

end do

write(11, \*, err = 100) (a(i), i = 1, n)

close(11)

close(12)

return

100 pause 'Error'

end

Программа перевода txt в bin

program main

implicit none

integer n, k

open(10, file = 'size.txt', err = 100)

read(10, \*, err = 100) n

close(10)

call txt2binint('ia.txt', 'ia.bin', n + 1)

open(10, file = 'ia.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

read(10, rec = n + 1, err = 100) k

call txt2binint('ja.txt', 'ja.bin', k - 1)

call txt2binreal('di.txt', 'di.bin', n)

call txt2binreal('al.txt' , 'al.bin', k - 1)

call txt2binreal('vector.txt', 'vector.bin', n)

return

100 pause 'error'

end

subroutine txt2binreal(txt, bin, n)

integer i, n

character \*(\*) txt, bin

real a

dimension a(100000)

open(11, file = txt, err = 100)

open(12, file = bin, form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

read(11, \*, err = 100) (a(i), i = 1, n)

do i = 1, n

write(12, rec = i, err = 100) a(i)

end do

close(11)

close(12)

return

100 pause 'error'

end

subroutine txt2binint(txt, bin, n)

integer i, n

character \*(\*) txt, bin

integer a

dimension a(100000)

open(11, file = txt, err = 100)

open(12, file = bin, form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

read(11, \*, err = 100) (a(i), i = 1, n)

do i = 1, n

write(12, rec = i, err = 100) a(i)

end do

close(11)

close(12)

return

100 pause 'Error'

end

Программа генерации матрицы и вектора большой размерности

program main

implicit none

integer \*4 n, m, step, max, i, j, str, col, pos

open(10, file = 'size.txt', err = 100)

read(10,\*,err=100) n, m

close(10)

open(11, file = 'ia.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

open(12, file = 'ja.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

open(13, file = 'di.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

open(14, file = 'al.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

open(15, file = 'vector.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

max = n\*(n-1)/2

if (m .gt. max) goto 100

do i = 1, n

write(13, rec = i, err = 100) 1.

write(15, rec = i, err = 100) 1.

end do

step = max / m

col = 1

pos = 1

str = 1

i = 1

write(11, rec = str, err = 100) pos

do str = 2, n

col = str - 1

write(11, rec = str, err = 100) pos

if (pos .gt. m) cycle

do j = i, col, step

write(14, rec = pos, err = 100) 1.

write(12, rec = pos, err = 100) j

pos = pos + 1

if (j .gt. col) exit

end do

i = str + i - col

end do

return

100 pause 'error'

end

Программа перевода матрицы из разреженного формата в плотный

program main

integer n, ia, ja

real al, di

dimension ia(100000)

dimension ja(100000), al(100000), di(100000)

open(10, file = 'size.txt', err = 100)

read (10, \*) n

open(11, file = 'ia.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

open(12, file = 'ja.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

open(13, file = 'di.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

open(14, file = 'al.bin', form = 'unformatted',

,access = 'direct', recl = 4, err = 100)

do i = 1, n

read (11, rec = i) ia(i)

read (13, rec = i) di(i)

end do

read(11, rec = n + 1) ia(n + 1)

do i = 1, ia(n + 1) - 1

read (12, rec = i) ja(i)

read (14, rec = i) al(i)

end do

call denseMatrix(ia(1), di(1), ja(1), al(1),n, ia(n + 1) - 1)

return

100 pause 'error'

end

subroutine denseMatrix(ia,di,ja,al,n,c)

integer i,j,n,c,ia,ja

real di,al,part

dimension ia(n+1),ja(c),di(n),al(c)

open(2,file='matrix.txt')

do i=1,n

do j=1,n

if(i .eq. j) then

write(2,'(f4.2,a1,\)') di(i),' '

else

if(i .gt. j) then

write(2,'(f4.2,a1,\)') part(j,i,ia,

,ja,al,n,c),' '

else

write(2,'(f4.2,a1,\)') part(i,j,ia,

,ja,al,n,c),' '

endif

endif

enddo

write(2,\*)

enddo

close(2)

end

function part(j,i,ia,ja,al,n,c)

integer i,j,k,l,n,c,ia,ja

real al,part

dimension ia(n+1),ja(c),al(c)

if(ia(i+1) .ne. ia(i)) then

k = ia(i+1) - ia(i)

do l=0,k-1

if(ja(ia(i)+l) .eq. j) then

part = al(ia(i)+l)

return

end if

end do

if(l .eq. k) part = 0.0

else

part = 0.0

end if

end

1. **Тесты**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Результат | Цель теста |
| 1 |  |  | Матрица с целыми числами |
| **2** |  |  | Матрица с вещественными числами |
| **3** |  |  | Сгенерированная матрица размером 10\*10 |

Результаты тестов совпали с ожидаемыми результатами.