Sparse Matrix

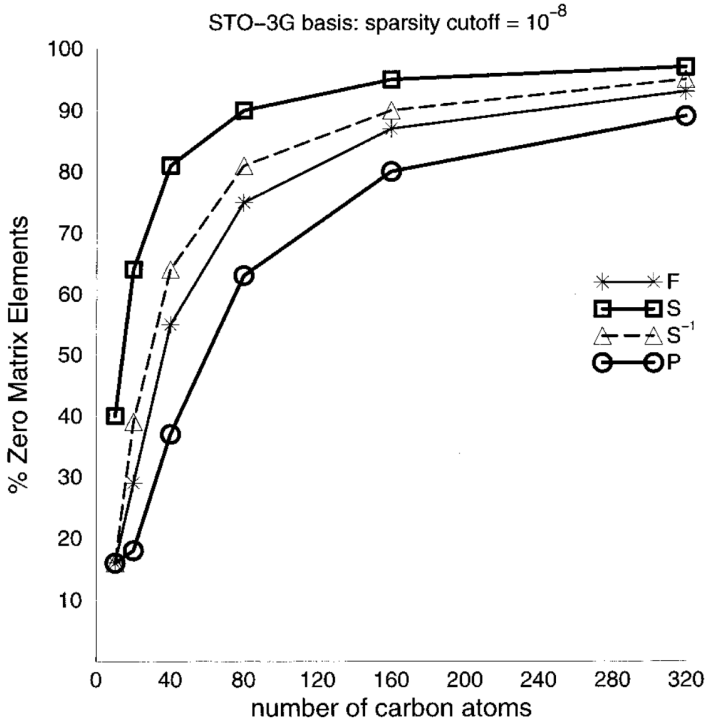
Yang Guo

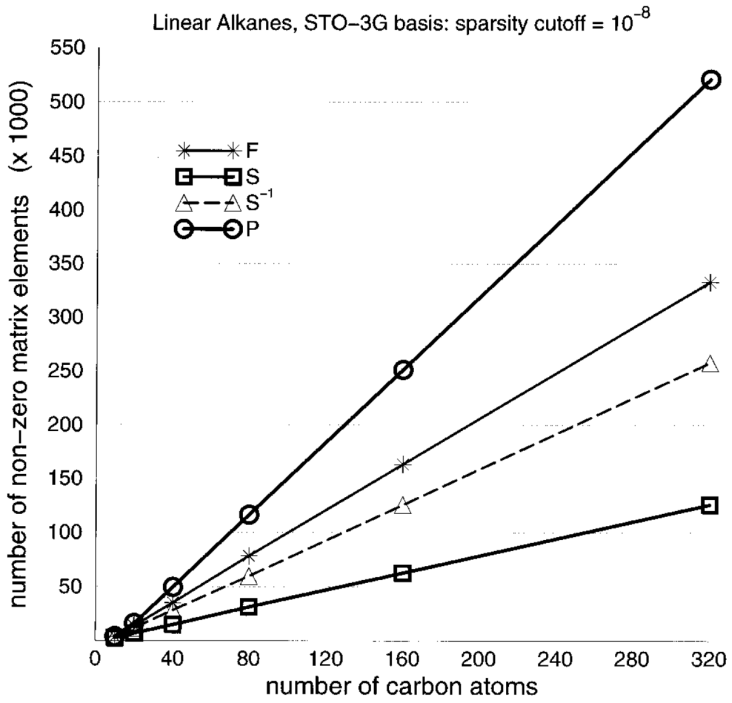
Catalog

- Background
- Various storage formats
- Matrix multiplication
- Difficulties

Sparsity

- Various matrix of 1D system





P. E. Maslen, C. Ochsenfeld, C. A. White, M. S. Lee, and M. Head-Gordon, J. Phys. Chem. A 1998, 102, 12, 2215

Sparse matrix storage

- Coordinate (COO) format

$$P=\begin{pmatrix} a_{11} & a_{12} & 0 & a_{14} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 & a_{26} \\ 0 & 0 & a_{33} & 0 & 0 & a_{36} \\ a_{41} & 0 & 0 & 0 & a_{45} & 0 \\ 0 & 0 & 0 & a_{54} & a_{55} & 0 \\ 0 & a_{62} & a_{63} & 0 & 0 & a_{66} \end{pmatrix}$$

Dim	6											15			
row index	1	1	1	2	2	2	3	3	4	4	5	5	6	6	6
column index	1	2	4	1	2	6	3	6	1	5	4	5	2	3	6
value array	a_{11}	a_{12}	a_{14}	a_{21}	a_{22}	a_{26}	a_{33}	a_{36}	a_{41}	a_{45}	a_{54}	a_{55}	a_{62}	a_{63}	a_{66}

- Compressed Sparse Row (CSR) format

$$P=\begin{pmatrix} a_{11} & a_{12} & 0 & a_{14} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 & a_{26} \\ 0 & 0 & a_{33} & 0 & 0 & a_{36} \\ a_{41} & 0 & 0 & 0 & a_{45} & 0 \\ 0 & 0 & 0 & a_{54} & a_{55} & 0 \\ 0 & a_{62} & a_{63} & 0 & 0 & a_{66} \end{pmatrix}$$

pionter array	1			4			7		9		11		13			16
	↓			↓			↓		↓		↓		↓			↓
column array	1	2	4	1	2	6	3	6	1	5	4	5	2	3	6	*End*
value array	a_{11}	a_{12}	a_{14}	a_{21}	a_{22}	a_{26}	a_{33}	a_{36}	a_{41}	a_{45}	a_{54}	a_{55}	a_{62}	a_{63}	a_{66}	*End*

Matrix to CSR

- Dry run before allocation
- Allocate the two integral and one real matrixes
- Matrix to CSR

```
subroutine CSR_Dim(Mat,n,m,n0,thre)
implicit none
integer::n,m,n0
real(kind=8) Mat(n,m),thre
integer::i,j

n0=0
do i=1,n
  do j=1,m
    if(dabs(Mat(i,j))>=thre) then
      n0=n0+1
    end if
  end do
end do

return
end
```

```
subroutine Mat_to_CSR(Mat,n,m,Va,Ja,Ia,thre,io)
implicit none
integer::n,m,io,Ja(*),Ia(n+1)
real(kind=8) Mat(n,m),Va(*),thre
integer::i,j,KK


KK=0;Ia(:)=0
Ia(1)=1
do i=1,n
  do j=1,m
    if(dabs(Mat(i,j))>=thre) then
      KK=KK+1
      Va(KK)=Mat(i,j)
      Ja(KK)=j
    end if
  end do
  Ia(i+1)=KK+1
end do

return
end
```

https://github.com/guoyang0123/QM_coding.git

Matrix multiplication

```
subroutine amub(nrow, ncol, a, ja, ia, b, jb, ib, c, jc, ic)
  use iso_fortran_env
  real(kind=8), intent(in) :: a(*)
  integer, intent(in) :: ja(*)
  integer, intent(in) :: ia(*)
  real(kind=8), intent(in) :: b(*)
  integer, intent(in) :: jb(*)
  integer, intent(in) :: ib(*)
  real(kind=8), intent(out) :: c(*)
  integer, intent(out) :: jc(*)
  integer, intent(out) :: ic(*)
  integer, allocatable :: iw(:)
  allocate(iw(ncol))
  len = 0
  ic(1) = 1
  iw(:) = 0
  do ii = 1, nrow
    do ka = ia(ii), ia(ii + 1) - 1
      scal = a(ka)
      jj = ja(ka)
      do kb = ib(jj), ib(jj + 1) - 1
        jcol = jb(kb)
        jpos = iw(jcol)
        if (jpos == 0) then
          len = len + 1
          jc(len) = jcol
          iw(jcol) = len
          c(len) = scal * b(kb)
        else
          c(jpos) = c(jpos) + scal * b(kb)
        end if
      end do
    end do
    do k = ic(ii), len
      iw(jc(k)) = 0
    end do
    ic(ii + 1) = len + 1
  end do
  deallocate(iw)
end subroutine amub
```

$$C_{ii,kk}=A_{ii,jj}*B_{jj,kk}$$

- The length of C is unknown before multiplication. As the input of amub, the dimensions of c should be M*N.
- The result C could consist of elements less than threshold, and should be trimmed.
- Similar subroutine could be used in the Davidson diagonalization driver involving sparse Hamiltonian matrixes (σ=HC)

https://github.com/guoyang0123/QM_coding.git

Homework

- Generate a 10000*10000 matrix **A**, with 90%, 10%, 1%, 0.1% non-zero matrix elements (Thresh=1D-6).
- Convert the above matrixes to CSR format, and perform the matrix multiplication **B=A*A**. Compare the computational time with **dgemm**, check the sparsity of B matrix.
- Write matrix multiplication program for **B=A*A** using **dgemm**, and compare the computational time with that with CSR format.

https://github.com/guoyang0123/QM_coding

GNU Project debugger

- Compile codes with gdb parameters, like “-DDEBUG -g”
- Launch “gdb”
- Execute the program within gdb environment
- Some keywords
  - backtrace
  - frame
  - print
  - …..

GNU Project debugger

- make clean
- make debug
- gdb
- file *.exe, run

```
(base) guo@LAPTOP-6K2BSI4T:~/QM_coding$ make clean
removed './build/objects/src/program.o'
removed './build/objects/src/tools/linear.o'
removed directory './build/objects/src/tools'
removed directory './build/objects/src'
removed './build/apps/Mat.x'
removed './build/apps/my_mat.dat'
(base) guo@LAPTOP-6K2BSI4T:~/QM_coding$ make debug
g++ -pedantic-errors -Wall -Wextra -DDEBUG -g -Iinclude/ -I/opt/intel/mkl/include  -o build/objects/src/tools/linear.o -c src/tools/linear.cpp
src/tools/linear.cpp: In function ‘void free_dvector(double*, int, int)’:
src/tools/linear.cpp:24:40: warning: unused parameter ‘nh’ [-Wunused-parameter]
   24 | void free_dvector(double *v,int nl,int nh)
      |                                    ~~~~^~
src/tools/linear.cpp: In function ‘void free_dmatrix(double**, int, int, int, int)’:
src/tools/linear.cpp:72:58: warning: unused parameter ‘nch’ [-Wunused-parameter]
   72 | void free_dmatrix(double **m,int nrl,int nrh,int ncl,int nch)
      |                                                      ~~~~^~~
g++ -pedantic-errors -Wall -Wextra -DDEBUG -g -Iinclude/ -I/opt/intel/mkl/include  -o build/objects/src/program.o -c src/program.cpp
g++ -pedantic-errors -Wall -Wextra -DDEBUG -g -Iinclude/ -I/opt/intel/mkl/include  -L/usr/lib -lstdc++ -DEIGEN_USE_MKL_ALL -L/opt/intel/mkl/lib/intel64 -L/opt/intel/lib/intel64 -Wl,--no-as-n
eeded -lmkl_rt -lpthread -lm -ldl -o ./build/apps/Mat.x ./build/objects/src/tools/linear.o ./build/objects/src/program.o
(base) guo@LAPTOP-6K2BSI4T:~/QM_coding$ gdb
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) file build/apps/Mat.x
Reading symbols from build/apps/Mat.x...
(gdb) run
```

GNU Project debugger

- backtrace
- frame #
- print
- q (quit)

```
(gdb) run
Starting program: /home/guo/QM_coding/build/apps/Mat.x
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Welcome to QM_coding!
There is no argument

  0.680375 -0.0452059  -0.967399  0.0258648    0.05349  0.0519907   -0.52344  0.0632129   0.375723   0.912937
 -0.211234   0.257742  -0.514226   0.678224   0.539828  -0.827888   0.941268  -0.921439  -0.668052    0.17728
  0.566198  -0.270431  -0.725537    0.22528  -0.199543  -0.615572   0.804416  -0.124725  -0.119791   0.314608
   0.59688  0.0268018   0.608354  -0.407937   0.783059   0.326454    0.70184    0.86367    0.76015   0.717353
  0.823295   0.904459  -0.686642   0.275105  -0.433371   0.780465  -0.466669    0.86162   0.658402  -0.12088
 -0.604897    0.83239  -0.198111  0.0485744  -0.295083  -0.302214  0.0795207   0.441905  -0.339326   0.84794
 -0.329554   0.271423  -0.740419  -0.012834   0.615449  -0.871657  -0.249586  -0.431413  -0.542064  -0.203127
  0.536459   0.434594  -0.782382    0.94555   0.838053  -0.959954   0.520497   0.477069   0.786745   0.629534
 -0.444451  -0.716795   0.997849  -0.414966  -0.860489 -0.0845965  0.0250707   0.279958   -0.29928   0.368437
   0.10794   0.213938  -0.563486   0.542715   0.898654  -0.873808   0.335448  -0.291903    0.37334   0.821944
Mat.x: include/Eigen/src/Core/DenseCoeffsBase.h:366: Eigen::DenseCoeffsBase<Derived, 1>::Scalar& Eigen::DenseCoeffsBase<Derived, 1>::operator()(Eigen::Index, Eigen::Index) [with Derived = E
gen::Matrix<double, -1, -1>; Eigen::DenseCoeffsBase<Derived, 1>::Scalar = double; Eigen::Index = long int]: Assertion `row >= 0 && row < rows() && col >= 0 && col < cols()' failed.

Program received signal SIGABRT, Aborted.
__GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
50      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) backtrace
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
#1  0x00007ffff92c5859 in __GI_abort () at abort.c:79
#2  0x00007ffff92c5729 in __assert_fail_base (fmt=0x7ffff945b588 "%s%s%s:%u: %s%sAssertion `%s' failed.\n%n", assertion=0x8007298 "row >= 0 && row < rows() && col >= 0 && col < cols()",
    file=0x8007268 "include/Eigen/src/Core/DenseCoeffsBase.h", line=366, function=<optimized out>) at assert.c:92
#3  0x00007ffff92d6f36 in __GI___assert_fail (assertion=0x8007298 "row >= 0 && row < rows() && col >= 0 && col < cols()", file=0x8007268 "include/Eigen/src/Core/DenseCoeffsBase.h",
    line=366,
    function=0x8007170 "Eigen::DenseCoeffsBase<Derived, 1>::Scalar& Eigen::DenseCoeffsBase<Derived, 1>::operator()(Eigen::Index, Eigen::Index) [with Derived = Eigen::Matrix<double, -1, -1>;
Eigen::DenseCoeffsBase<Derived, 1>"...) at assert.c:101
#4  0x0000000008003c3b in Eigen::DenseCoeffsBase<Eigen::Matrix<double, -1, -1, 0, -1, -1>, 1>::operator() (this=0x7fffffedae0, row=10, col=0)
    at include/Eigen/src/Core/DenseCoeffsBase.h:366
#5  0x0000000008003360 in Retrieve_Matrix (f=0x801f260, matrix=...) at src/tools/linear.cpp:257
#6  0x0000000008004e80 in main (argc=1, argv=0x7fffffedd18) at src/program.cpp:53
(gdb) frame 5
#5  0x0000000008003360 in Retrieve_Matrix (f=0x801f260, matrix=...) at src/tools/linear.cpp:257
257                 matrix(i,j)=vec[k];
(gdb) print j
$1 = 0
(gdb) print i
$2 = 10
```