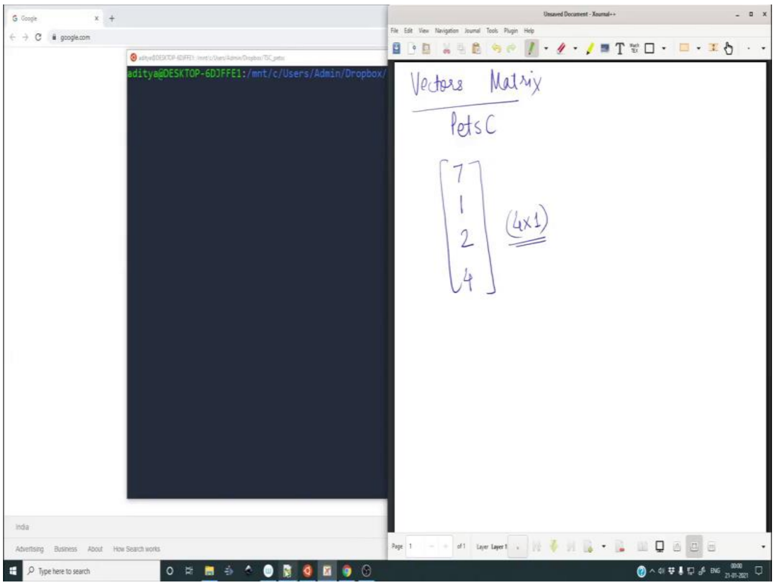
Tools in Scientific Computing
Prof. Aditya Bandopadhyay
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 30
PETSc – Creating Vectors and Matrices

(Refer Slide Time: 00:27)

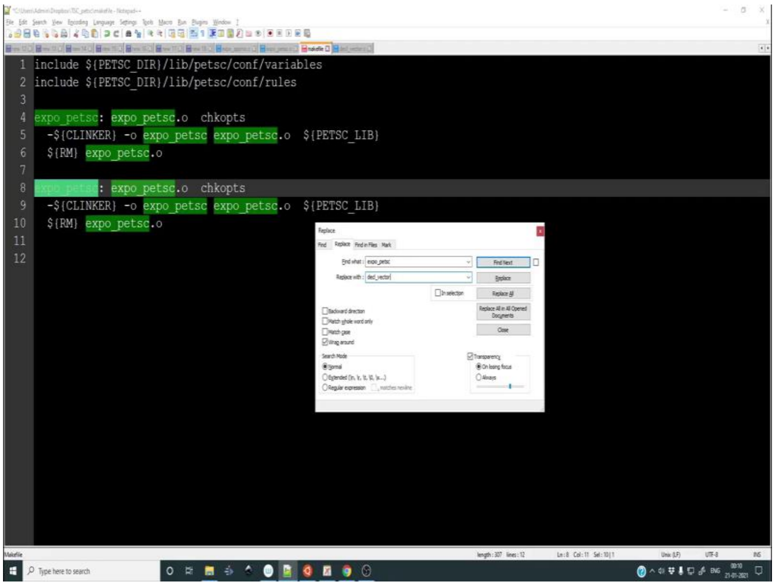


```
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/
```

Hi everyone, in this particular lecture we are going to look at matrices and vectors in Pets C. And if time permitting we are going to solve a simple algebraic equation and see all the objects that we can access. So, in the previous class we looked at some aspects of m p i, but in this particular lecture we are going to make use of a single processor just to avoid any kind of confusion.

So, let us try to create this particular vector. [7 1 2 4]; so it is a 4 × 1 vector 4 rows and 1 column and yeah; so, all the elements are non zero as you can see. So, let us see how we can go up go about defining this particular vector.

(Refer Slide Time: 01:45)



```
include ${PETSC_DIR}/lib/petsc/conf/variables
include ${PETSC_DIR}/lib/petsc/conf/rules

expo_petsc: expo_petsc.o  chkopts
	-${CLINKER} -o expo_petsc expo_petsc.o  ${PETSC_LIB}
	${RM} expo_petsc.o

expo_petsc: expo_petsc.o  chkopts
	-${CLINKER} -o expo_petsc expo_petsc.o  ${PETSC_LIB}
	${RM} expo_petsc.o
```

Replace

Find what : expo_petsc

Replace with : decl_vector

□ In selection

□ Backward direction

□ Match whole word only

□ Match case

☑ Wrap around

Search Mode

● Normal

○ Extended (\n, \r, \t, \0, \x...)

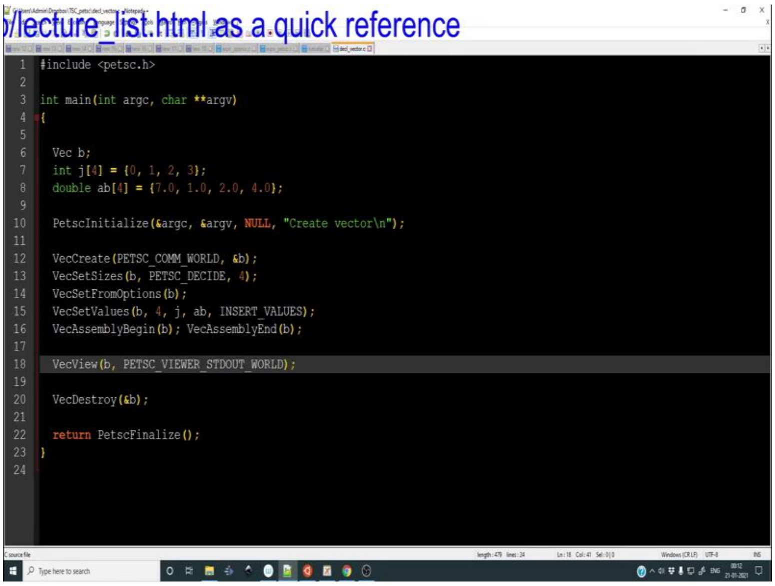
○ Regular expression

☑ Transparency

● On losing focus

○ Always

(Refer Slide Time: 01:57)



```c
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Create vector\n");

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES);
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);

  return PetscFinalize();
}
```

So, as usual we will create a new file and first things, first we are going to import petsc. So, hash include std not std petsc.h. Then we are going to make the main function. So, int main arg sorry int argc char star star argv and we are going to return the output of petsc finalize.

We have looked at petsc finalize. So, petsc finalize its wrapping up the code and the return value is simply the i e r r that is the error signal. So, it’s going to be an integer. So, we can return that that is the conclusion of the code. So, there are certain things that every petsc code must have right.

So, first things first there has to be a Petsc initialize. If there is a finalize, there has to be an initialize as well and the initialize takes the command line arguments and some help string alright. So, with this out of the way so we have petsc initialize and this.

So, now we must declare certain variables. So, in order to declare an array, the way to do it is to tell which elements are going to be non zero right. So, what we can do is; so, let us declare some element, some variables that we will need. So, we will need a vector, let us call it b. So, vec is a data type inside petsc and so it is a vector right. So, similar to this there are there is a data type called mat, but we will come to that later.

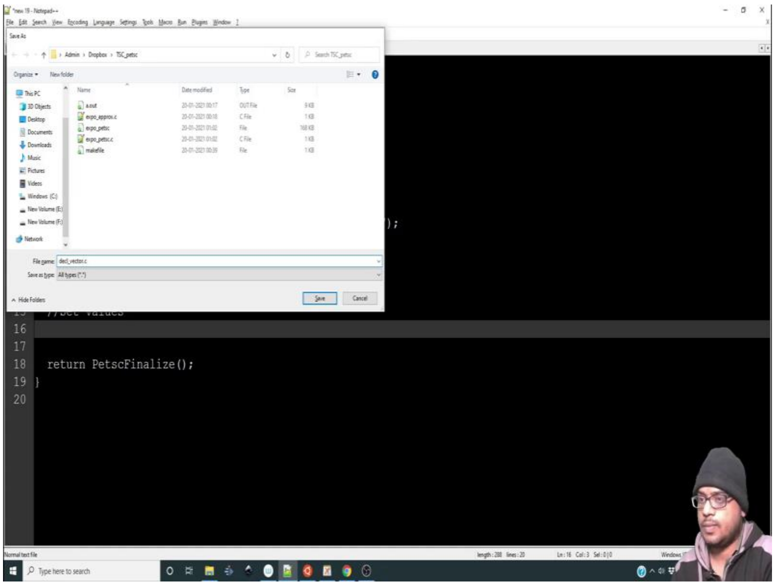
So, b is a vector alright. So, we will need the integers where we are going to insert values and of course, we are going to need some will be, we are going to need an array of numbers with which we are going to fill in the array b ok.

So, I mean this is a very synthetic example and over here you are not going to truly appreciate the power of petsc, but I am going to give you some pointers on how this may work out, not may work out it works out.

So, we are going to create a double array we are going to call it a b just to distinguish it from b and the a prefix I use it so that it reminds me that it is like an auxiliary array to b ok. So, let me declare this as so what did we have? 7 1 2 4. So, 7.0 1.0 2.0 and 4.0. So, this is the elements that we want to put inside b and apart from this what else are we want to need anyway. So, as we go ahead we are going to declare things as and when they come ok.

So, the way to initialize or create a vector is the following. So, first things first you must do VecCreate, you must then do VecSetSizes. So, this is a common procedure of declaring a vector. Then we must do VecSet from sizes from options, then we must so over here we must set values ok, let me save this file alright.

(Refer Slide Time: 06:20)



Name	Date modified	Type	Size
a.out	20-01-2021 00:17	OUT File	9 KB
expo_approx.c	20-01-2021 00:18	C File	1 KB
expo_petsc	20-01-2021 01:02	File	168 KB
expo_petsc.c	20-01-2021 01:02	C File	1 KB
makefile	20-01-2021 00:39	File	1 KB

File name: decl_vector.c

Save as type: All types (*.*)

```
  //Set values


  return PetscFinalize();
}
```

So, now we have some colors. So, there must be some code over here which will sort of assign values into this, then what do we have we must then assemble the vector. So, VecAssemblyBegin and then we must follow this particular function called by VecAssemblyEnd and it is just a idiosyncrasy of this particular library, you have to do this always.

This these two follow each other. So, I have not yet put in the arguments of this function calls, but the argument of assembly end is going to be the vector that you are going to populate and b is also going to be this; so yeah so far so good.

So, now VecCreate; so VecCreate must be done over. So, since we are using one processor. Well, it is not just a question of one processor; it is a question of making the address of b available to all the threads that you will declare. That is why you have PETSC COMM WORLD and you are giving the address of b. Now, inside set size you must tell that we must set the size for b and we must let petsc decide I will come to the syntax in a while and the size is going to be 4 alright.

So, petsc decide, this particular keyword is used to signify the fact that if you are using multiple processors so, one processor will maybe initialize these two, one processor will initialize these two. Things like this, I mean so for such a small array it does not matter right, but once it starts becoming very big, you may benefit tremendously by distributing the load over several processors alright.

So, now once we have set the size of b. So, this is the overall size and petsc decide will decide how to distribute the load alright. So now, set from options it has to be only passed to b. Now, we must set values. So, there are various ways of setting values. So, let us start with method one.

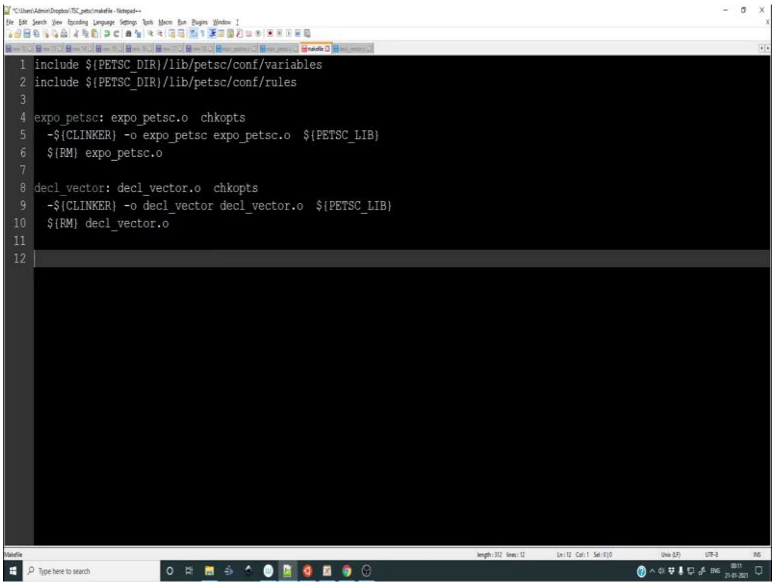
I mean rather than going with method one I will I am going to show you the easiest method which is VecSet values. And VecSet values must first accept the argument b that is the vector you want to populate, then how many entries do you want to populate that is; 4, then we must specify the indices where you want to populate.

So, we must create an integer array. So, let me call it j and we will make this j 4 and we will make it 0, 1, 2, 3 right. So, j is the address of the array which holds the indices where we want to insert these nonzero entries and I will change this and show you how it will affect the code.

So, we want to insert 4 values at the j locations. So, j is the indices where you want to insert and lastly, we want to tell what exactly we want to insert. So, we want to insert the entries inside the array a b. So, a b is the address of the array 7, 1, 2, 4 and how do you want to set the values? There are two modes; one is insert values and one is add values.

So, if there is an already existing entry for b, if you do add values you will sort of add whatever you are doing to the existing values. If you do insert values that will erase the value insert the present values alright; so far so good. So, yeah this assembles the matrix and in the end before finalizing we must do vec destroys, we must clear the variables. So, we must give the address of b and that is it. So, let us save this, let us go to our make file and create a new target.

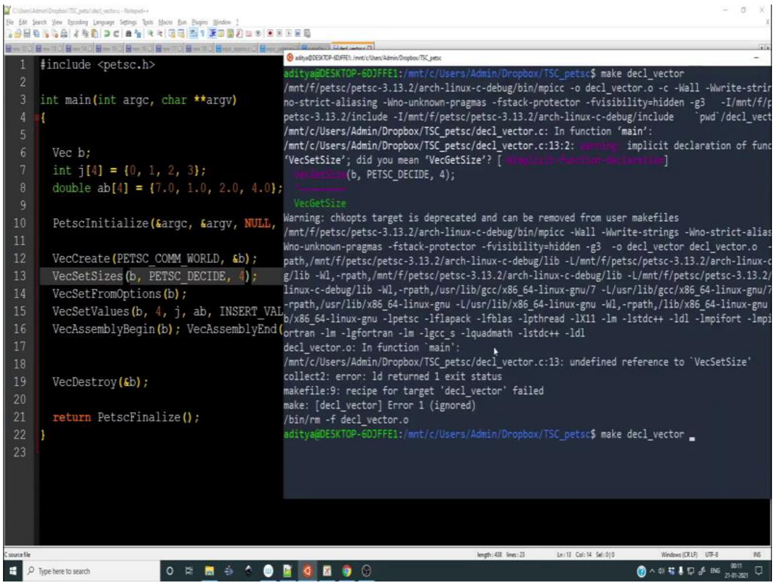
(Refer Slide Time: 11:34)



```
include ${PETSC_DIR}/lib/petsc/conf/variables
include ${PETSC_DIR}/lib/petsc/conf/rules

expo_petsc: expo_petsc.o  chkopts
	-${CLINKER} -o expo_petsc expo_petsc.o  ${PETSC_LIB}
	${RM} expo_petsc.o

decl_vector: decl_vector.o  chkopts
	-${CLINKER} -o decl_vector decl_vector.o  ${PETSC_LIB}
	${RM} decl_vector.o

```

So, let me hit conrol H oops, control H and I want to replace this by declare vector ok. So, replace, replace, replace, replace, replace. So, there is no use of like rewriting this entire thing. We simply having the same variables and rules. We are just making a new target right.

(Refer Slide Time: 11:54)

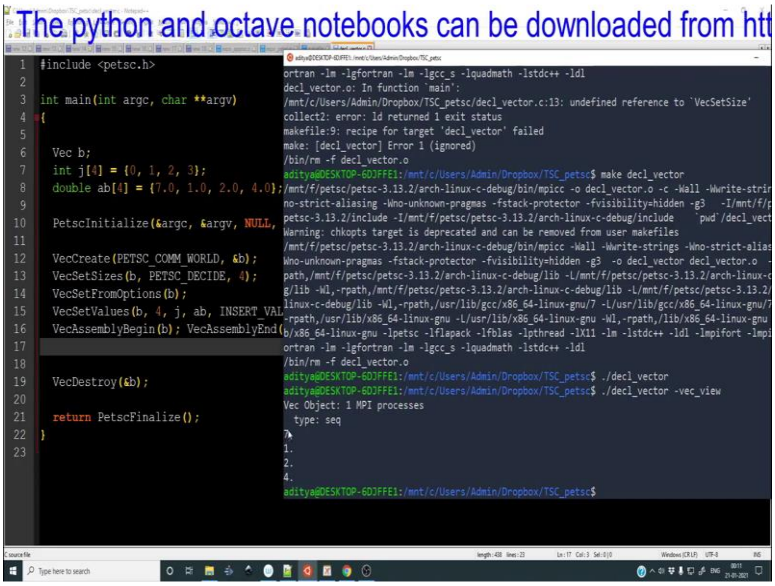


```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(


  VecDestroy(&b);

  return PetscFinalize();
}

```

```
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strir
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c: In function ‘main’:
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:13:2: warning: implicit declaration of func
‘VecSetSize’; did you mean ‘VecGetSize’? [-Wimplicit-function-declaration]
  VecSetSize(b, PETSC_DECIDE, 4);
  ^~~~~~~~~~
  VecGetSize
Warning: chkopts target is deprecated and can be removed from user makefiles
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
decl_vector.o: In function `main':
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:13: undefined reference to `VecSetSize'
collect2: error: ld returned 1 exit status
makefile:9: recipe for target 'decl_vector' failed
make: [decl_vector] Error 1 (ignored)
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
```

So, let us save this file. So, we have our code over here alright. So, make decl vector ok, there appears to be an error. So, let us see what the error is. So, there appears to be so, this has to be set sizes oops! Yeah, ok.

(Refer Slide Time: 12:23)

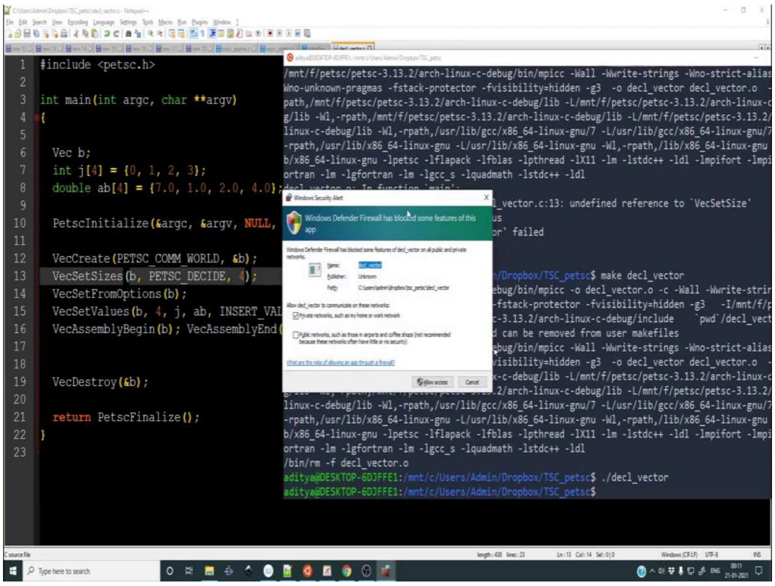


The python and octave notebooks can be downloaded from htt

```c
#include <petsc.h>

int main(int argc, char **argv)
{


  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(


  VecDestroy(&b);

  return PetscFinalize();
}
```

```
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
decl_vector.o: In function `main':
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:13: undefined reference to `VecSetSize'
collect2: error: ld returned 1 exit status
makefile:9: recipe for target 'decl_vector' failed
make: [decl_vector] Error 1 (ignored)
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strin
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
Warning: chkopts target is deprecated and can be removed from user makefiles
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -vec_view
Vec Object: 1 MPI processes
  type: seq
7.
1.
2.
4.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

(Refer Slide Time: 12:27)



```c
#include <petsc.h>

int main(int argc, char **argv)
{


  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(


  VecDestroy(&b);

  return PetscFinalize();
}
```

```
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
                                      l_vector.c:13: undefined reference to `VecSetSize'
                                      us
                                      or' failed


                                      /Dropbox/TSC_petsc$ make decl_vector
                                      ebug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strin
                                      -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
                                      -3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
                                      d can be removed from user makefiles
                                      ebug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
                                      visibility=hidden -g3  -o decl_vector decl_vector.o  -
                                      x-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
                                      .2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

Windows Security Alert

Windows Defender Firewall has blocked some features of this app

Windows Defender Firewall has blocked some features of decl_vector on all public and private networks.

Name: decl_vector
Publisher: Unknown
Path: C:\users\admin\dropbox\tsc_petsc\decl_vector

Allow decl_vector to communicate on these networks:

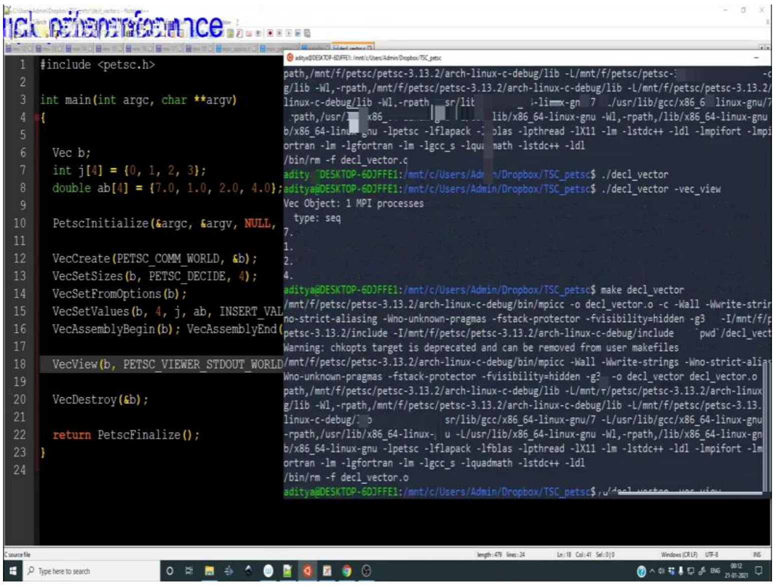
☑ Private networks, such as my home or work network

☐ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

So, it compiles nice and well. So, let us just do dot slash declare vector we have to allow this ok. So, it does nothing as expected, because we have not printed anything we have not like asked it to do anything for us. But what we can do is pass this command line argument minus VecVeiw and it will show us the elements of the vector ok.

So, additionally we can insert this VecVeiw inside the code itself. So, what we can do is vec so, before destroying the vector we can do VecVeiw, we will pass b, then petsc viewer std out world, just to say that it has to run on a single process.

(Refer Slide Time: 13:23)



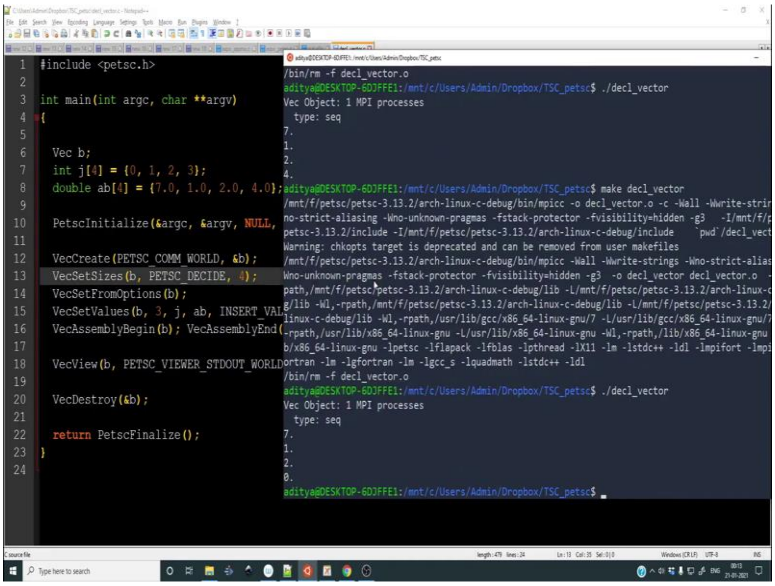
```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_STDOUT_WORLD

  VecDestroy(&b);

  return PetscFinalize();
}
```

```
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-                -c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath, sr/lit          -linux-gn 7  ./usr/lib/gcc/x86_6 linux-gnu/7
 path,/usr/  x86_              lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-lin  gnu -lpetsc -lflapack - blas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lqua math -lstdc++ -ldl
/bin/rm -f decl_vector.c
adity DESKTOP-6DJFFE1:/mnt/c/Users/Adm n/Dropbox/TSC_petsc$ ./decl_vector
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -vec_view
Vec Object: 1 MPI processes
  type: seq
7.
1.
2.
4.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strir
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
Warning: chkopts target is deprecated and can be removed from user makefiles
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alia
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.
linux-c-debug/  o              sr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu
-rpath,/usr/lib/x86_64-linux- u -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gn
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lm
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

So, let us make it again, let us run it and now, it shows without having to pass the extra argument. So, without using this ok, alright so, we have now initialized this now. So, what happens? If we still make j equal to this, but we only do b as so, we only want to insert 3 values.

So, we still have the 4 indices and the 4 values, but instead we say we want to insert only 3 first indices and the first three indices the first three values. Let us see what happens.

(Refer Slide Time: 14:12)

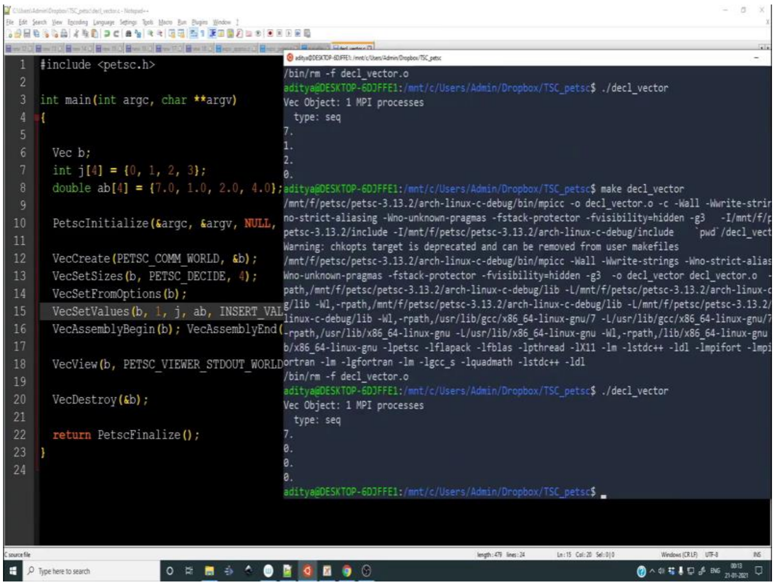


```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 3, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_STDOUT_WORLD

  VecDestroy(&b);

  return PetscFinalize();
}
```

```
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
Vec Object: 1 MPI processes
  type: seq
7.
1.
2.
4.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strir
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
Warning: chkopts target is deprecated and can be removed from user makefiles
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
Vec Object: 1 MPI processes
  type: seq
7.
1.
2.
0.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

So, look its initialized it to 0 instead of getting a 4, because we have not told it to populate b with 4 values, we only told it to populate with 3 values.

(Refer Slide Time: 14:31)

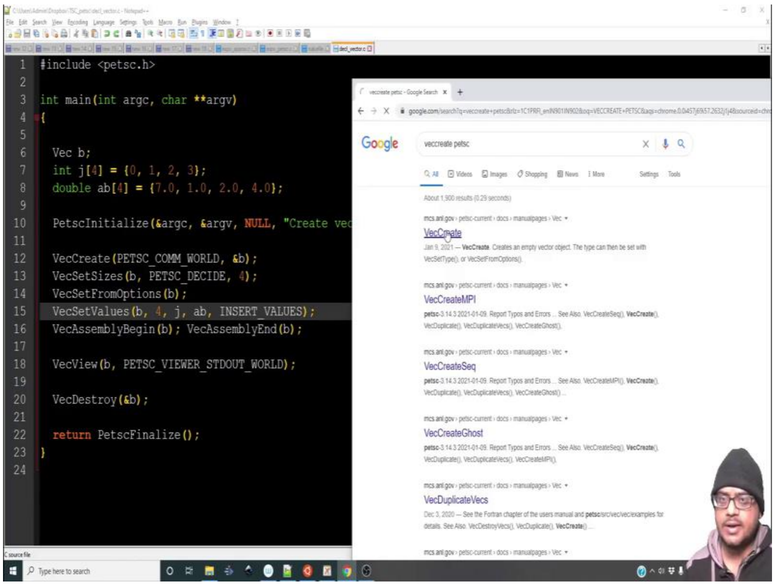


```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 1, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_STDOUT_WORLD

  VecDestroy(&b);

  return PetscFinalize();
}
```

```
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
Vec Object: 1 MPI processes
  type: seq
7.
1.
2.
0.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strir
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
Warning: chkopts target is deprecated and can be removed from user makefiles
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
Vec Object: 1 MPI processes
  type: seq
7.
0.
0.
0.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ _
```

Similarly, if I do only one it will only populate the first value. Let us make it again value ok. So, it is incredibly versatile you can declare everything you want, but in the end if you want to fill partially you can choose to do that, there is nothing wrong in that. So, this is how we can declare vectors and sort of print them out. Well, there are various ways of viewing a vector as well.

(Refer Slide Time: 15:03)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Create vec

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES);
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);

  return PetscFinalize();
}
```

veccreate petsc - Google Search
google.com/search?q=veccreate+petsc&rlz=1C1PRFI_enIN901IN902&oq=VECCREATE+PETSC&aqs=chrome.0.0i457j69i57.2632j1j4&sourceid=chr
Google
veccreate petsc
All
Videos
Images
Shopping
News
More
Settings
Tools

About 1,900 results (0.29 seconds)

mcs.anl.gov › petsc-current › docs › manualpages › Vec
VecCreate
Jan 9, 2021 — VecCreate. Creates an empty vector object. The type can then be set with VecSetType(), or VecSetFromOptions().

mcs.anl.gov › petsc-current › docs › manualpages › Vec
VecCreateMPI
petsc-3.14.3 2021-01-09. Report Typos and Errors ... See Also. VecCreateSeq(), VecCreate(), VecDuplicate(), VecDuplicateVecs(), VecCreateGhost().

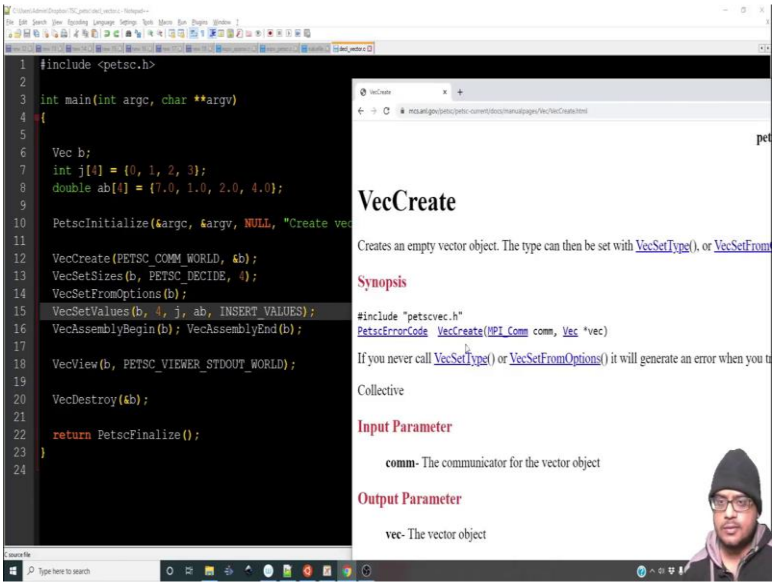
mcs.anl.gov › petsc-current › docs › manualpages › Vec
VecCreateSeq
petsc-3.14.3 2021-01-09. Report Typos and Errors ... See Also. VecCreateMPI(), VecCreate(), VecDuplicate(), VecDuplicateVecs(), VecCreateGhost() ...

mcs.anl.gov › petsc-current › docs › manualpages › Vec
VecCreateGhost
petsc-3.14.3 2021-01-09. Report Typos and Errors ... See Also. VecCreateSeq(), VecCreate(), VecDuplicate(), VecDuplicateVecs(), VecCreateMPI().

mcs.anl.gov › petsc-current › docs › manualpages › Vec
VecDuplicateVecs
Dec 3, 2020 — See the Fortran chapter of the users manual and petsc/src/vec/vec/examples for details. See Also. VecDestroyVecs(), VecDuplicate(), VecCreate() ...

mcs.anl.gov › petsc-current › docs › manualpages › Vec

(Refer Slide Time: 15:14)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Create vec

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES);
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);

  return PetscFinalize();
}
```

VecCreate

Creates an empty vector object. The type can then be set with VecSetType(), or VecSetFrom

Synopsis

```
#include "petscvec.h"
PetscErrorCode  VecCreate(MPI_Comm comm, Vec *vec)
```

If you never call VecSetType() or VecSetFromOptions() it will generate an error when you t

Collective

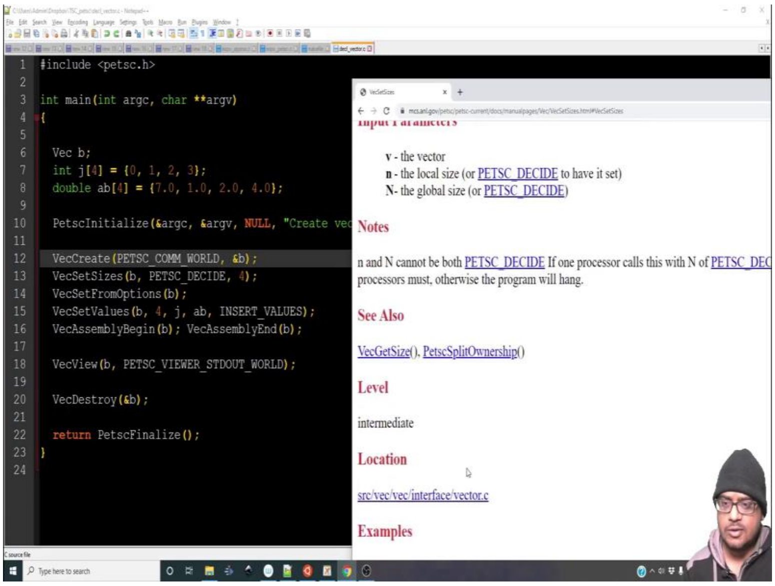
Input Parameter

comm- The communicator for the vector object

Output Parameter

vec- The vector object

(Refer Slide Time: 15:24)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Create vec

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES);
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);

  return PetscFinalize();
}
```

Input Parameters

v - the vector
n - the local size (or PETSC_DECIDE to have it set)
N- the global size (or PETSC_DECIDE)

Notes

n and N cannot be both PETSC_DECIDE If one processor calls this with N of PETSC_DEC
processors must, otherwise the program will hang.

See Also

VecGetSize(), PetscSplitOwnership()

Level

intermediate

Location

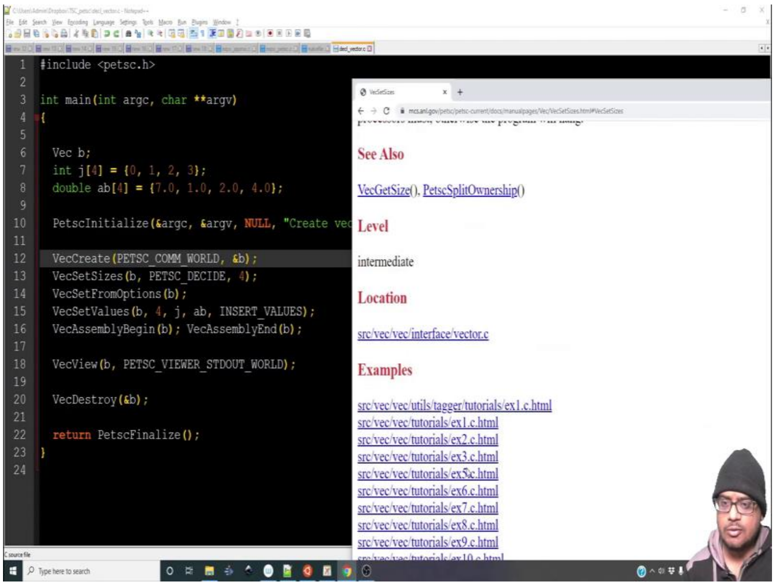
src/vec/vec/interface/vector.c

Examples

So, like I told I am going to show you the function reference. So, VecCreate so, it requires the MPI communicator which in this case is PETSC COMM WORLD and it requires the pointer to b the address of b. So, we have passed the address of b ampersand b that is standard c syntax. So, VecSetSizes so, it sets the local sizes.

So, we have to pass the vector the small n is the local size if you have multiple processors then you have to pass small n, but in this case we have let petsc decide and then the global size alright. So, the global size of our array is 4, but we have asked petsc to decide for us the best load balancing it can do. In this case we are running it on one processor. So, it does not matter anyway alright. So, with these options it is going to initialize b over here.

(Refer Slide Time: 16:18)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Create vec

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES);
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);

  return PetscFinalize();
}
```

See Also

VecGetSize(), PetscSplitOwnership()

Level

intermediate

Location

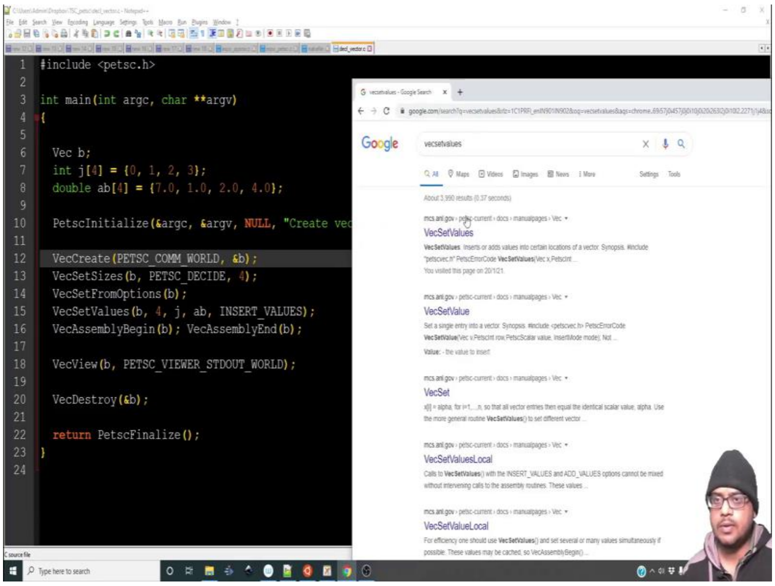
src/vec/vec/interface/vector.c

Examples

src/vec/vec/utils/tagger/tutorials/ex1.c.html
src/vec/vec/tutorials/ex1.c.html
src/vec/vec/tutorials/ex2.c.html
src/vec/vec/tutorials/ex3.c.html
src/vec/vec/tutorials/ex5c.html
src/vec/vec/tutorials/ex6.c.html
src/vec/vec/tutorials/ex7.c.html
src/vec/vec/tutorials/ex8.c.html
src/vec/vec/tutorials/ex9.c.html

So, now VecSet Values let me go to the function reference like set values alright.

(Refer Slide Time: 16:29)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Create vec

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES);
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);

  return PetscFinalize();
}
```

vecsetvalues

About 3,390 results (0.37 seconds)

mcs.anl.gov › petsc-current › docs › manualpages › Vec
VecSetValues
VecSetValues. Inserts or adds values into certain locations of a vector. Synopsis. #include "petscvec.h" PetscErrorCode VecSetValues(Vec x,PetscInt ...
You visited this page on 20/1/21.

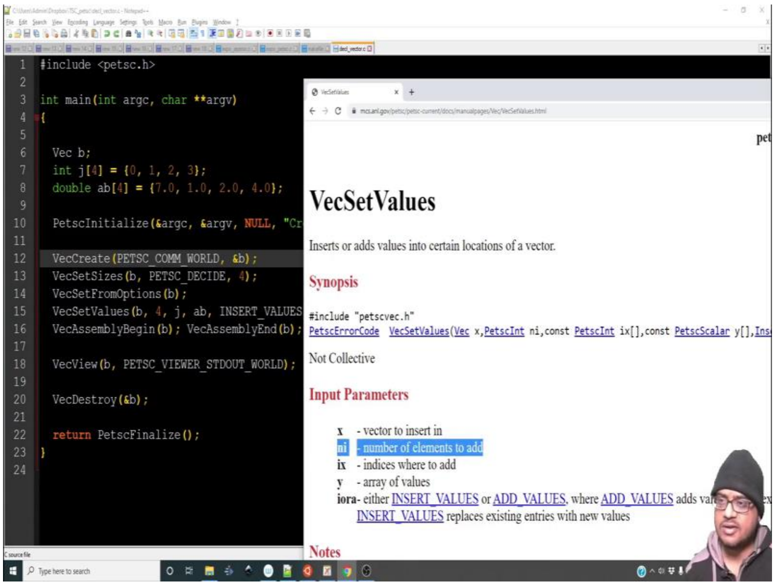
mcs.anl.gov › petsc-current › docs › manualpages › Vec
VecSetValue
Set a single entry into a vector. Synopsis. #include <petscvec.h> PetscErrorCode VecSetValue(Vec v,PetscInt row,PetscScalar value, InsertMode mode); Not ...
Value: - the value to insert

mcs.anl.gov › petsc-current › docs › manualpages › Vec
VecSet
x[i] = alpha, for i=1,...,n, so that all vector entries then equal the identical scalar value, alpha. Use the more general routine VecSetValues() to set different vector ...

mcs.anl.gov › petsc-current › docs › manualpages › Vec
VecSetValuesLocal
Calls to VecSetValues() with the INSERT_VALUES and ADD_VALUES options cannot be mixed without intervening calls to the assembly routines. These values ...

mcs.anl.gov › petsc-current › docs › manualpages › Vec
VecSetValueLocal
For efficiency one should use VecSetValues() and set several or many values simultaneously if possible. These values may be cached, so VecAssemblyBegin() ...

(Refer Slide Time: 16:31)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Cr

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);

  return PetscFinalize();
}
```

mcs.anl.gov/petsc/petsc-current/docs/manualpages/Vec/VecSetValues.html

VecSetValues

Inserts or adds values into certain locations of a vector.

Synopsis

```
#include "petscvec.h"
PetscErrorCode  VecSetValues(Vec x,PetscInt ni,const PetscInt ix[],const PetscScalar y[],Ins
```

Not Collective

Input Parameters

x - vector to insert in
ni - number of elements to add
ix - indices where to add
y - array of values
iora- either INSERT_VALUES or ADD_VALUES, where ADD_VALUES adds va
INSERT_VALUES replaces existing entries with new values

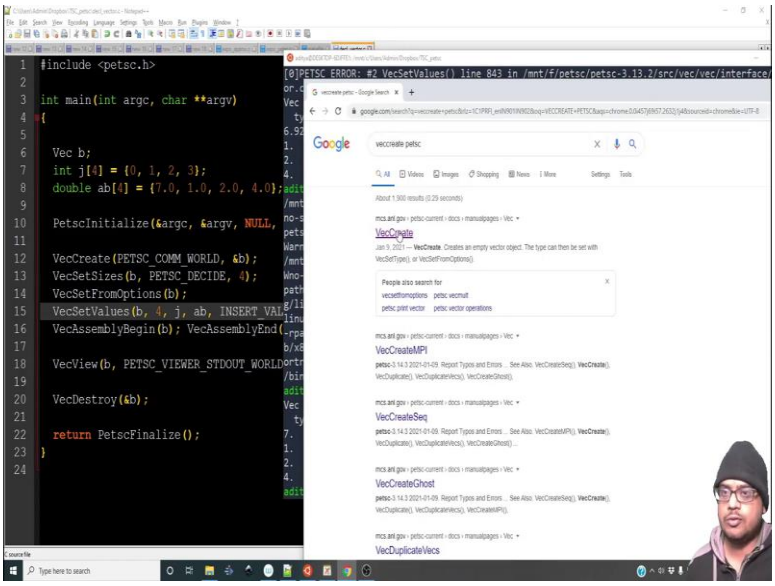
Notes

So, it requires the vector the number of elements to add look is the it is number of elements right. In this case we wanted 4 and I have showed you what happens when you reduce it.

If you increase it, it will let us see what happens when we increase it ok. Before that we must show the, we must tell what the indices of insertion are ok and we must tell what values you want to insert and you can either insert values or add values. So, it is quite simple. This is the standard syntax of creating a vector in petsc.

So, now let us increase this and see what happens let me make it 6 just to go beyond the boundary case. So, let me make this; obviously, there is not going to be an error over here, because we would not done any illegal sentence and there is an error argument out of range. So, it cannot do that, it gives a big garbage value. So, we must have this maximum 4. So, let me fix it great. So, now we will proceed to figure out how we can declare matrices in this particular way.

(Refer Slide Time: 17:57)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_STDOUT_WORLD

  VecDestroy(&b);

  return PetscFinalize();
}
```

```
[0]PETSC ERROR: #2 VecSetValues() line 843 in /mnt/f/petsc/petsc-3.13.2/src/vec/vec/interface/
```

veccreate petsc

About 1,900 results (0.29 seconds)

mcs.anl.gov › petsc-current › docs › manualpages › Vec

VecCreate

Jan 9, 2021 — VecCreate. Creates an empty vector object. The type can then be set with VecSetType(), or VecSetFromOptions().

People also search for

vecsetfromoptions petsc vecmult

petsc print vector petsc vector operations

mcs.anl.gov › petsc-current › docs › manualpages › Vec

VecCreateMPI

petsc-3.14.3 2021-01-09. Report Typos and Errors ... See Also. VecCreateSeq(), VecCreate(), VecDuplicate(), VecDuplicateVecs(), VecCreateGhost().

mcs.anl.gov › petsc-current › docs › manualpages › Vec

VecCreateSeq

petsc-3.14.3 2021-01-09. Report Typos and Errors ... See Also. VecCreateMPI(), VecCreate(), VecDuplicate(), VecDuplicateVecs(), VecCreateGhost() ...

mcs.anl.gov › petsc-current › docs › manualpages › Vec

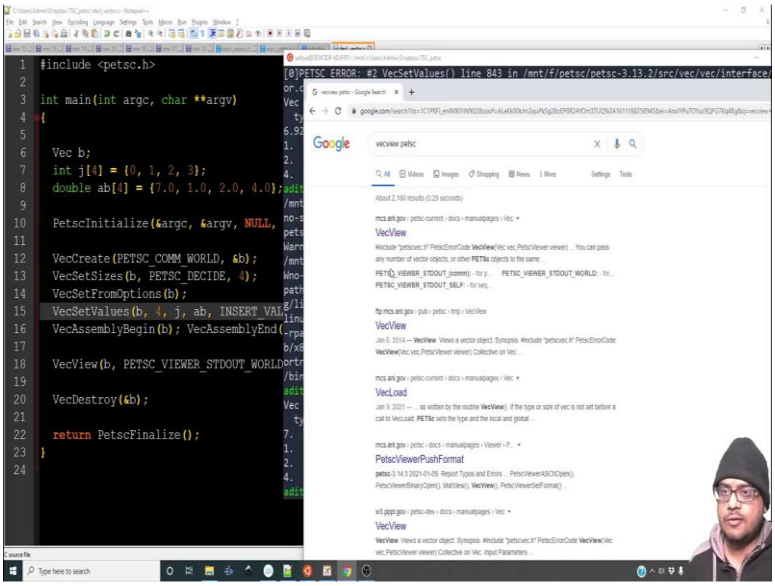
VecCreateGhost

petsc-3.14.3 2021-01-09. Report Typos and Errors ... See Also. VecCreateSeq(), VecCreate(), VecDuplicate(), VecDuplicateVecs(), VecCreateMPI().

mcs.anl.gov › petsc-current › docs › manualpages › Vec

VecDuplicateVecs

(Refer Slide Time: 18:05)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_STDOUT_WORLD

  VecDestroy(&b);

  return PetscFinalize();
}
```

```
[0]PETSC ERROR: #2 VecSetValues() line 843 in /mnt/f/petsc/petsc-3.13.2/src/vec/vec/interface/
```

vecview petsc

About 2,180 results (0.29 seconds)

mcs.anl.gov › petsc-current › docs › manualpages › Vec

VecView

#include "petscvec.h" PetscErrorCode VecView(Vec vec,PetscViewer viewer) ... You can pass any number of vector objects, or other PETSc objects to the same ...

PETSC_VIEWER_STDOUT_(comm): - for p... PETSC_VIEWER_STDOUT_WORLD: - for...

PETSC_VIEWER_STDOUT_SELF: - for seq...

ftp.mcs.anl.gov › pub › petsc › tmp › VecView

VecView

Jan 6, 2014 — VecView. Views a vector object. Synopsis. #include "petscvec.h" PetscErrorCode VecView(Vec vec,PetscViewer viewer) Collective on Vec ...

mcs.anl.gov › petsc-current › docs › manualpages › Vec

VecLoad

Jan 9, 2021 — ... as written by the routine VecView(). If the type or size of vec is not set before a call to VecLoad, PETSc sets the type and the local and global ...

mcs.anl.gov › petsc › docs › manualpages › Viewer › P...

PetscViewerPushFormat

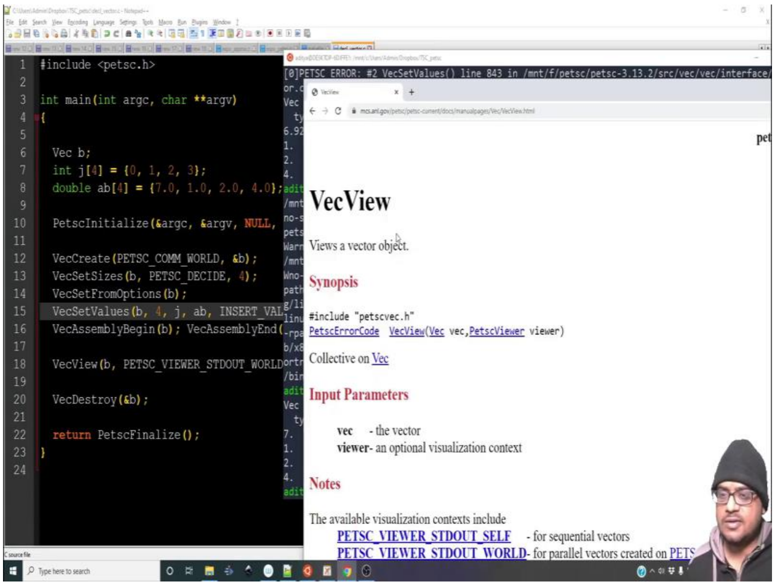
petsc-3.14.3 2021-01-09. Report Typos and Errors ... PetscViewerASCIIOpen(), PetscViewerBinaryOpen(), MatView(), VecView(), PetscViewerSetFormat() ...

w3.pppl.gov › petsc-dev › docs › manualpages › Vec

VecView

VecView. Views a vector object. Synopsis. #include "petscvec.h" PetscErrorCode VecView(Vec vec,PetscViewer viewer) Collective on Vec. Input Parameters ...

(Refer Slide Time: 18:06)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_STDOUT_WORLD

  VecDestroy(&b);

  return PetscFinalize();
}
```

```
[0]PETSC ERROR: #2 VecSetValues() line 843 in /mnt/f/petsc/petsc-3.13.2/src/vec/vec/interface/
```

mcs.anl.gov/petsc/petsc-current/docs/manualpages/Vec/VecView.html

VecView

Views a vector object.

Synopsis

```
#include "petscvec.h"
PetscErrorCode  VecView(Vec vec,PetscViewer viewer)
```

Collective on Vec

Input Parameters

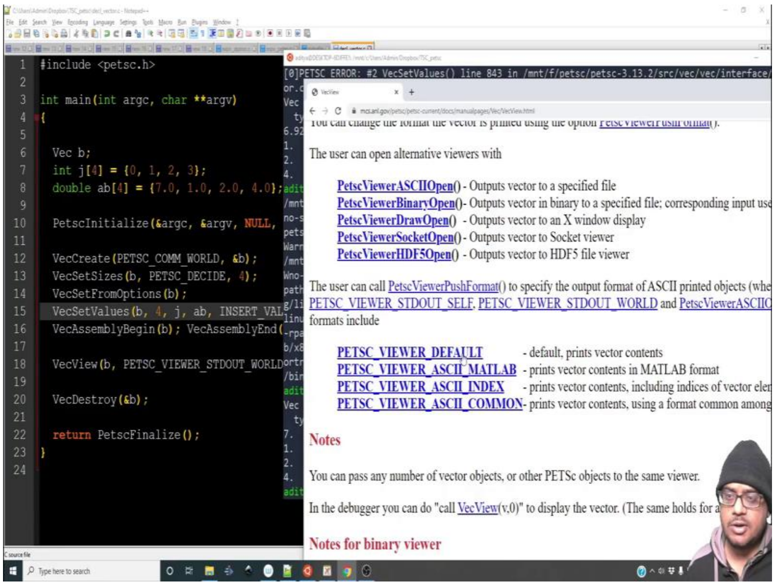
vec - the vector
viewer- an optional visualization context

Notes

The available visualization contexts include

PETSC_VIEWER_STDOUT_SELF - for sequential vectors
PETSC_VIEWER_STDOUT_WORLD- for parallel vectors created on PETS

(Refer Slide Time: 18:09)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_STDOUT_WORLD

  VecDestroy(&b);

  return PetscFinalize();
}
```

```
[0]PETSC ERROR: #2 VecSetValues() line 843 in /mnt/f/petsc/petsc-3.13.2/src/vec/vec/interface/
```

You can change the format the vector is printed using the option PetscViewerPushFormat().

The user can open alternative viewers with

PetscViewerASCIIOpen() - Outputs vector to a specified file
PetscViewerBinaryOpen()- Outputs vector in binary to a specified file; corresponding input use
PetscViewerDrawOpen() - Outputs vector to an X window display
PetscViewerSocketOpen() - Outputs vector to Socket viewer
PetscViewerHDF5Open() - Outputs vector to HDF5 file viewer

The user can call PetscViewerPushFormat() to specify the output format of ASCII printed objects (whe PETSC_VIEWER_STDOUT_SELF, PETSC_VIEWER_STDOUT_WORLD and PetscViewerASCIIO formats include

PETSC_VIEWER_DEFAULT - default, prints vector contents
PETSC_VIEWER_ASCII_MATLAB - prints vector contents in MATLAB format
PETSC_VIEWER_ASCII_INDEX - prints vector contents, including indices of vector eler
PETSC_VIEWER_ASCII_COMMON- prints vector contents, using a format common among

Notes

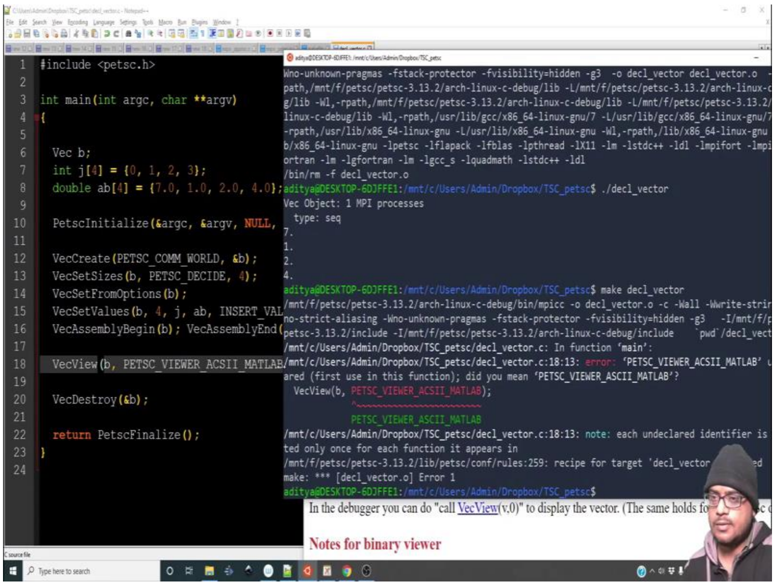
You can pass any number of vector objects, or other PETSc objects to the same viewer.

In the debugger you can do "call VecView(v,0)" to display the vector. (The same holds for a

Notes for binary viewer

But before we go to that let us just go to VecView, because I want to show you function reference for VecVeiw. There are various kinds of viewers. So, this ASCII, binary, draw ok. So, there are various kinds of viewers that we have. So, this a PETSC VIEWER DEFAULT, PETSC VIEWER ASCII MATLAB. Let us see what happens when we do that.

(Refer Slide Time: 18:40)



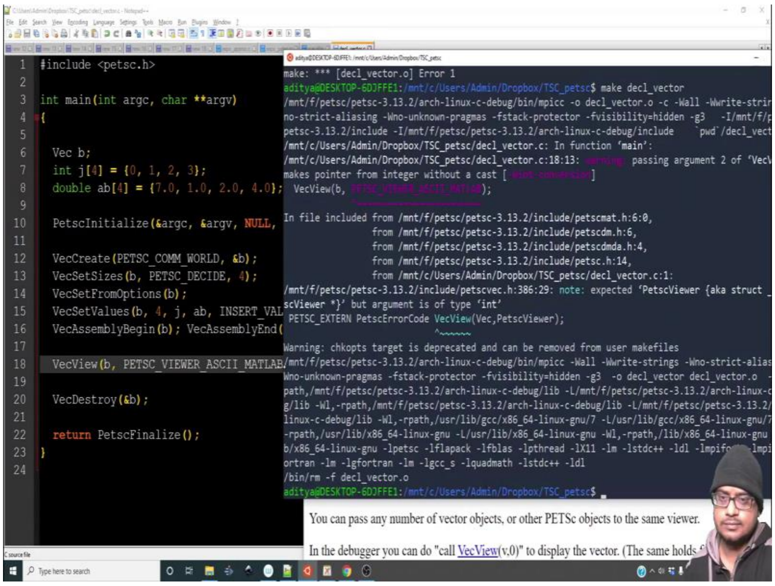
```c
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_ACSII_MATLAB

  VecDestroy(&b);

  return PetscFinalize();
}
```

```
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
Vec Object: 1 MPI processes
  type: seq
7.
1.
2.
4.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strir
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c: In function 'main':
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:18:13: error: 'PETSC_VIEWER_ACSII_MATLAB' u
ared (first use in this function); did you mean 'PETSC_VIEWER_ASCII_MATLAB'?
  VecView(b, PETSC_VIEWER_ACSII_MATLAB);
             ^~~~~~~~~~~~~~~~~~~~~~~~~
             PETSC_VIEWER_ASCII_MATLAB
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:18:13: note: each undeclared identifier is
ted only once for each function it appears in
/mnt/f/petsc/petsc-3.13.2/lib/petsc/conf/rules:259: recipe for target 'decl_vector        ed
make: *** [decl_vector.o] Error 1
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

In the debugger you can do "call VecView(v,0)" to display the vector. (The same holds fo

Notes for binary viewer

(Refer Slide Time: 18:50)



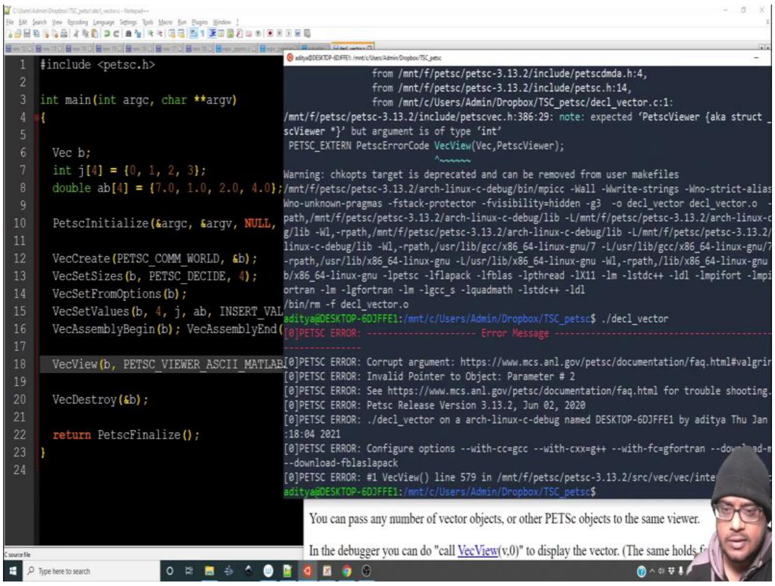
```c
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_ASCII_MATLAB

  VecDestroy(&b);

  return PetscFinalize();
}
```

```
make: *** [decl_vector.o] Error 1
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strir
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c: In function 'main':
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:18:13: warning: passing argument 2 of 'VecV
makes pointer from integer without a cast [-Wint-conversion]
  VecView(b, PETSC_VIEWER_ASCII_MATLAB);
             ^~~~~~~~~~~~~~~~~~~~~~~~~

In file included from /mnt/f/petsc/petsc-3.13.2/include/petscmat.h:6:0,
                 from /mnt/f/petsc/petsc-3.13.2/include/petscdm.h:6,
                 from /mnt/f/petsc/petsc-3.13.2/include/petscdmda.h:4,
                 from /mnt/f/petsc/petsc-3.13.2/include/petsc.h:14,
                 from /mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:1:
/mnt/f/petsc/petsc-3.13.2/include/petscvec.h:386:29: note: expected 'PetscViewer {aka struct _
scViewer *}' but argument is of type 'int'
 PETSC_EXTERN PetscErrorCode VecView(Vec,PetscViewer);
                             ^~~~~~~

Warning: chkopts target is deprecated and can be removed from user makefiles
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifo   -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ _
```

You can pass any number of vector objects, or other PETSc objects to the same viewer.

In the debugger you can do "call VecView(v,0)" to display the vector. (The same holds f

(Refer Slide Time: 18:56)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_ASCII_MATLAB

  VecDestroy(&b);

  return PetscFinalize();
}
```

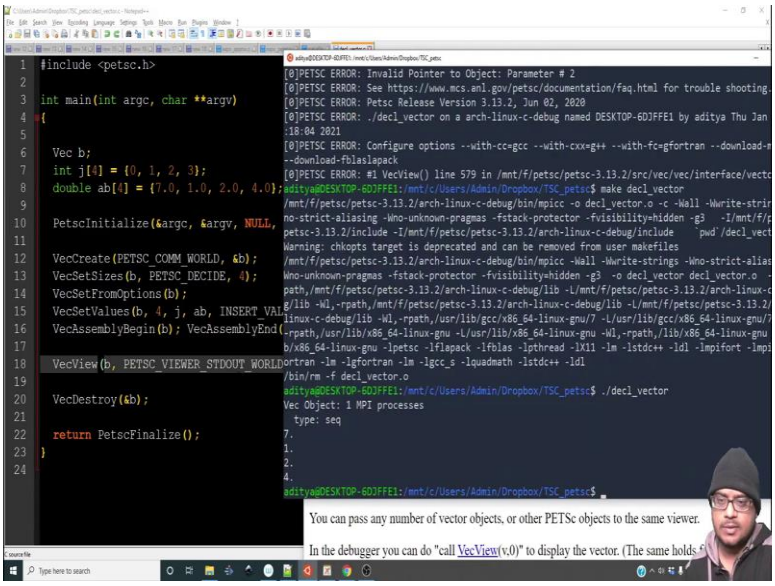
```
                 from /mnt/f/petsc/petsc-3.13.2/include/petscdmda.h:4,
                 from /mnt/f/petsc/petsc-3.13.2/include/petsc.h:14,
                 from /mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:1:
/mnt/f/petsc/petsc-3.13.2/include/petscvec.h:386:29: note: expected ‘PetscViewer {aka struct _
scViewer *}’ but argument is of type ‘int’
 PETSC_EXTERN PetscErrorCode VecView(Vec,PetscViewer);
                             ^~~~~~~
Warning: chkopts target is deprecated and can be removed from user makefiles
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
[0]PETSC ERROR: --------------------- Error Message ------------------------------------------
--------------
[0]PETSC ERROR: Corrupt argument: https://www.mcs.anl.gov/petsc/documentation/faq.html#valgrin
[0]PETSC ERROR: Invalid Pointer to Object: Parameter # 2
[0]PETSC ERROR: See https://www.mcs.anl.gov/petsc/documentation/faq.html for trouble shooting.
[0]PETSC ERROR: Petsc Release Version 3.13.2, Jun 02, 2020
[0]PETSC ERROR: ./decl_vector on a arch-linux-c-debug named DESKTOP-6DJFFE1 by aditya Thu Jan
:18:04 2021
[0]PETSC ERROR: Configure options --with-cc=gcc --with-cxx=g++ --with-fc=gfortran --download-m
--download-fblaslapack
[0]PETSC ERROR: #1 VecView() line 579 in /mnt/f/petsc/petsc-3.13.2/src/vec/vec/inte
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

You can pass any number of vector objects, or other PETSc objects to the same viewer.

In the debugger you can do "call VecView(v,0)" to display the vector. (The same holds f

It will show us the output in MATLAB forward. This is spelling mistake, ASCII alright. Well, there is an error and it does have to do something with the viewer.

(Refer Slide Time: 19:10)



```
#include <petsc.h>

int main(int argc, char **argv)
{

  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_STDOUT_WORLD

  VecDestroy(&b);

  return PetscFinalize();
}
```

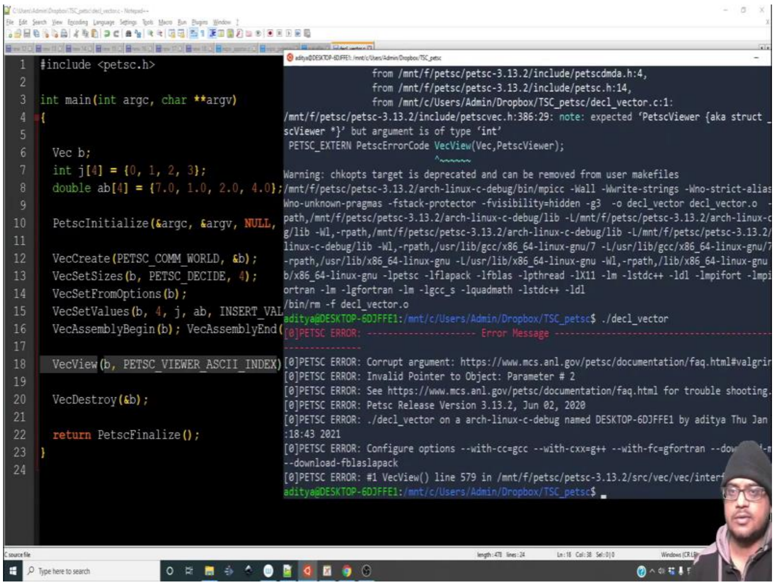
```
[0]PETSC ERROR: Invalid Pointer to Object: Parameter # 2
[0]PETSC ERROR: See https://www.mcs.anl.gov/petsc/documentation/faq.html for trouble shooting.
[0]PETSC ERROR: Petsc Release Version 3.13.2, Jun 02, 2020
[0]PETSC ERROR: ./decl_vector on a arch-linux-c-debug named DESKTOP-6DJFFE1 by aditya Thu Jan
:18:04 2021
[0]PETSC ERROR: Configure options --with-cc=gcc --with-cxx=g++ --with-fc=gfortran --download-m
--download-fblaslapack
[0]PETSC ERROR: #1 VecView() line 579 in /mnt/f/petsc/petsc-3.13.2/src/vec/vec/interface/vecto
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strin
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
Warning: chkopts target is deprecated and can be removed from user makefiles
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
Vec Object: 1 MPI processes
  type: seq
7.
1.
2.
4.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ _
```

You can pass any number of vector objects, or other PETSc objects to the same viewer.

In the debugger you can do "call VecView(v,0)" to display the vector. (The same holds f

Well, for some reason it does not work on my PC, but maybe it does on your ok. So, let us see whether this should work, there is no reason for this not to work.

(Refer Slide Time: 19:27)

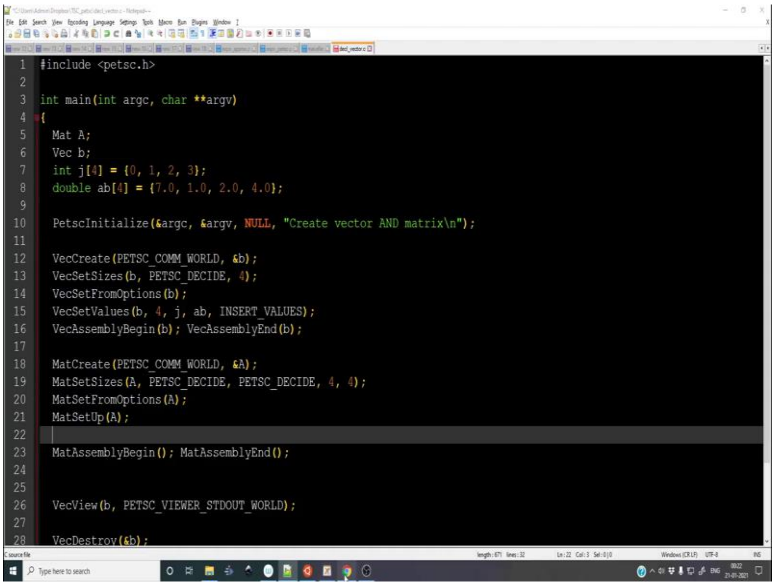


```c
#include <petsc.h>

int main(int argc, char **argv)
{


  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  VecView(b, PETSC_VIEWER_ASCII_INDEX)

  VecDestroy(&b);

  return PetscFinalize();
}
```

```
                 from /mnt/f/petsc/petsc-3.13.2/include/petscdmda.h:4,
                 from /mnt/f/petsc/petsc-3.13.2/include/petsc.h:14,
                 from /mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:1:
/mnt/f/petsc/petsc-3.13.2/include/petscvec.h:386:29: note: expected 'PetscViewer {aka struct _
scViewer *}' but argument is of type 'int'
 PETSC_EXTERN PetscErrorCode VecView(Vec,PetscViewer);
                             ^~~~~~~
Warning: chkopts target is deprecated and can be removed from user makefiles
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
[0]PETSC ERROR: --------------------- Error Message -----------------------------------------
--------------
[0]PETSC ERROR: Corrupt argument: https://www.mcs.anl.gov/petsc/documentation/faq.html#valgrir
[0]PETSC ERROR: Invalid Pointer to Object: Parameter # 2
[0]PETSC ERROR: See https://www.mcs.anl.gov/petsc/documentation/faq.html for trouble shooting.
[0]PETSC ERROR: Petsc Release Version 3.13.2, Jun 02, 2020
[0]PETSC ERROR: ./decl_vector on a arch-linux-c-debug named DESKTOP-6DJFFE1 by aditya Thu Jan
:18:43 2021
[0]PETSC ERROR: Configure options --with-cc=gcc --with-cxx=g++ --with-fc=gfortran --dow
--download-fblaslapack
[0]PETSC ERROR: #1 VecView() line 579 in /mnt/f/petsc/petsc-3.13.2/src/vec/vec/inter
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

Let us see, there is some errors that is weird ok. Anyway, ok. So, I need to call this one anyway. So, forget about this, do not worry about this, is as a beginner course. We will just keep it as STDOUT _WORLD and yeah. So, do not worry about that, we can call we can push a format and then we can bring it, but you do not need to worry about all that. So, so far we have this grid. So, now let us proceed to create a matrix alright so in fact, let us modify this file itself.

(Refer Slide Time: 20:35)



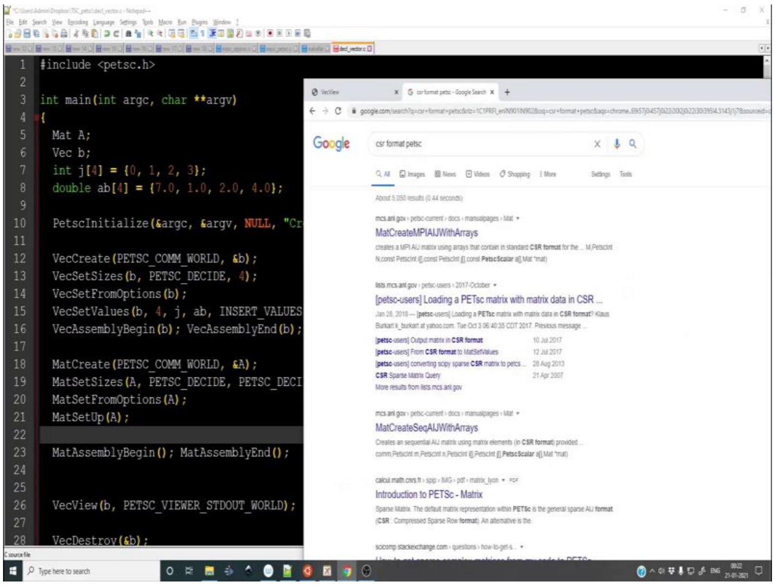
```c
#include <petsc.h>

int main(int argc, char **argv)
{
  Mat A;
  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Create vector AND matrix\n");

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES);
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  MatCreate(PETSC_COMM_WORLD, &A);
  MatSetSizes(A, PETSC_DECIDE, PETSC_DECIDE, 4, 4);
  MatSetFromOptions(A);
  MatSetUp(A);

  MatAssemblyBegin(); MatAssemblyEnd();


  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);
```

So, yeah similar to this there will be a sequence which will follow exactly this vector creation sequence. So, there will be a MatCreate, then there will be a MatSetSizes, then there will be a MatSet from options then we will have MatSetUp it is to and this is different from the vector, but you need to set up the matrix, then there has to be insertion of values lastly, there has to be an assembly and so, for that there will be two steps; so, MatAssemblyBegin followed by MatAssemblyEnd.

So, unlike the vector, the assembly has to have a certain key word additional, but first let us declare the matrix. So, Mat A alright. So, we have declared a matrix A and let us create the matrix A. So, A will also be 4 ×4 suppose. So, in order to create once again we have to pass this and the address of A that is common, set sizes again. So, for vector you need to only declare one dimension, but for matrices you need to declare two dimensions.

So, A, PETSC DECIDE comma PETS; PETSC DECIDE, because we have two dimensions and the processor load has to be balanced in two directions and the size of the matrix. It is pretty straightforward. Set from options it is simply going to have the variable A, after this we are going to set up A. Now, we have to insert values. Well, there are again various ways of inserting values, but the most common is the Compressed Storage Row format, the CSR format Compressed Sparse Row format ok.

(Refer Slide Time: 23:02)



```
#include <petsc.h>

int main(int argc, char **argv)
{
  Mat A;
  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Cr

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  MatCreate(PETSC_COMM_WORLD, &A);
  MatSetSizes(A, PETSC_DECIDE, PETSC_DECI
  MatSetFromOptions(A);
  MatSetUp(A);

  MatAssemblyBegin(); MatAssemblyEnd();


  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);
```

csr format petsc

About 5,050 results (0.44 seconds)

mcs.anl.gov › petsc-current › docs › manualpages › Mat
MatCreateMPIAIJWithArrays
creates a MPI AIJ matrix using arrays that contain in standard CSR format for the ... M,PetscInt N,const PetscInt i[],const PetscInt j[],const PetscScalar a[],Mat *mat)

lists.mcs.anl.gov › petsc-users › 2017-October
[petsc-users] Loading a PETsc matrix with matrix data in CSR ...
Jan 28, 2018 — [petsc-users] Loading a PETsc matrix with matrix data in CSR format? Klaus Burkart k_burkart at yahoo.com. Tue Oct 3 06:40:35 CDT 2017. Previous message ...

[petsc-users] Output matrix in CSR format	10 Jul 2017
[petsc-users] From CSR format to MatSetValues	12 Jul 2017
[petsc-users] converting scipy sparse CSR matrix to petcs ...	28 Aug 2013
CSR Sparse Matrix Query	21 Apr 2007

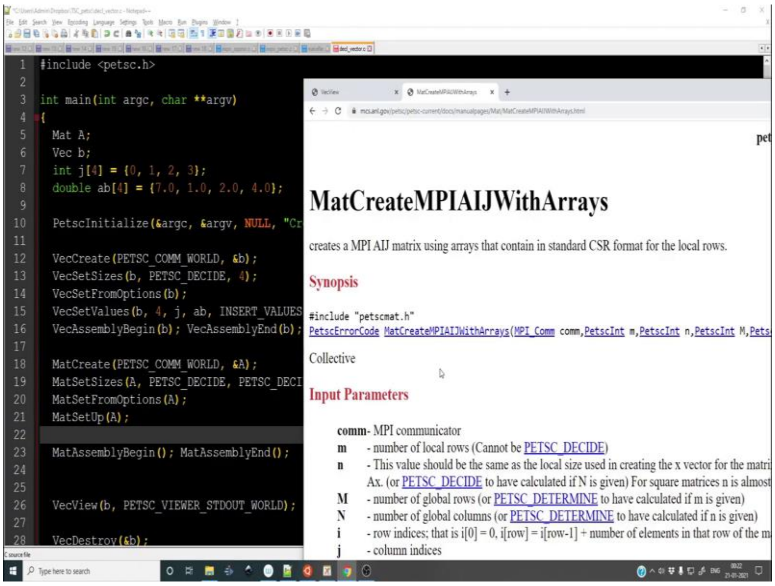
More results from lists.mcs.anl.gov

mcs.anl.gov › petsc-current › docs › manualpages › Mat
MatCreateSeqAIJWithArrays
Creates an sequential AIJ matrix using matrix elements (in CSR format) provided ... comm,PetscInt m,PetscInt n,PetscInt i[],PetscInt j[],PetscScalar a[],Mat *mat)

calcul.math.cnrs.fr › spip › IMG › pdf › matrix_lyon
Introduction to PETSc - Matrix
Sparse Matrix. The default matrix representation within PETSc is the general sparse AIJ format (CSR : Compressed Sparse Row format). An alternative is the

scicomp.stackexchange.com › questions › how-to-get-s...

(Refer Slide Time: 23:11)



```
#include <petsc.h>

int main(int argc, char **argv)
{
  Mat A;
  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Cr

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  MatCreate(PETSC_COMM_WORLD, &A);
  MatSetSizes(A, PETSC_DECIDE, PETSC_DECI
  MatSetFromOptions(A);
  MatSetUp(A);

  MatAssemblyBegin(); MatAssemblyEnd();


  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);
```

MatCreateMPIAIJWithArrays

creates a MPI AIJ matrix using arrays that contain in standard CSR format for the local rows.

Synopsis

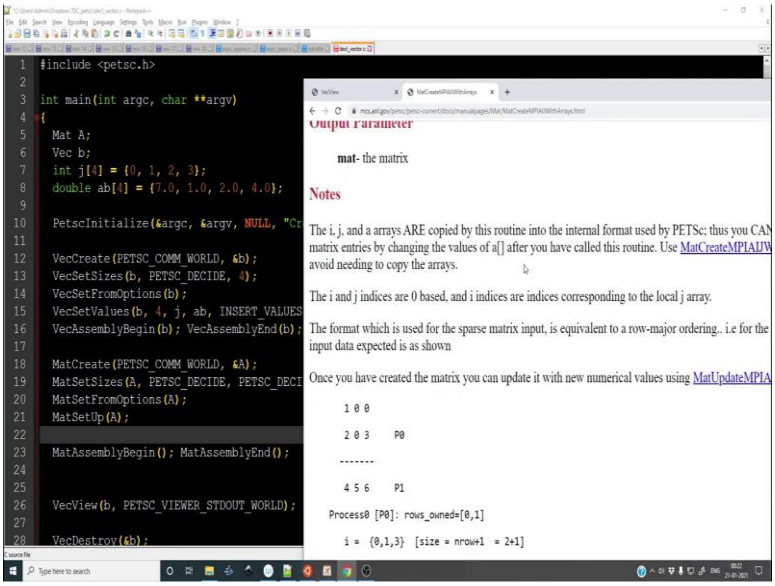
```
#include "petscmat.h"
PetscErrorCode MatCreateMPIAIJWithArrays(MPI_Comm comm,PetscInt m,PetscInt n,PetscInt M,Pets
```

Collective

Input Parameters

comm- MPI communicator
m - number of local rows (Cannot be PETSC_DECIDE)
n - This value should be the same as the local size used in creating the x vector for the matri Ax. (or PETSC_DECIDE to have calculated if N is given) For square matrices n is almost
M - number of global rows (or PETSC_DETERMINE to have calculated if m is given)
N - number of global columns (or PETSC_DETERMINE to have calculated if n is given)
i - row indices; that is i[0] = 0, i[row] = i[row-1] + number of elements in that row of the m
j - column indices

(Refer Slide Time: 23:20)



```
#include <petsc.h>

int main(int argc, char **argv)
{
  Mat A;
  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Cr

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  MatCreate(PETSC_COMM_WORLD, &A);
  MatSetSizes(A, PETSC_DECIDE, PETSC_DECI
  MatSetFromOptions(A);
  MatSetUp(A);

  MatAssemblyBegin(); MatAssemblyEnd();


  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);
```

Output Parameter

mat- the matrix

Notes

The i, j, and a arrays ARE copied by this routine into the internal format used by PETSc; thus you CAN matrix entries by changing the values of a[] after you have called this routine. Use MatCreateMPIAIJW avoid needing to copy the arrays.

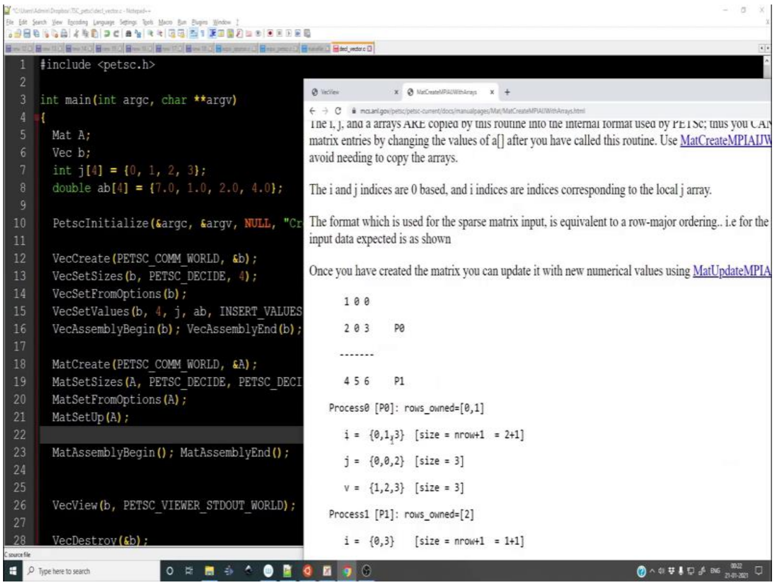
The i and j indices are 0 based, and i indices are indices corresponding to the local j array.

The format which is used for the sparse matrix input, is equivalent to a row-major ordering.. i.e for the input data expected is as shown

Once you have created the matrix you can update it with new numerical values using MatUpdateMPIA

```
    1 0 0

    2 0 3     P0

   -------

    4 5 6     P1

  Process0 [P0]: rows_owned=[0,1]

    i =  {0,1,3}  [size = nrow+1  = 2+1]
```

(Refer Slide Time: 23:23)



```
#include <petsc.h>

int main(int argc, char **argv)
{
  Mat A;
  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL, "Cr

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  MatCreate(PETSC_COMM_WORLD, &A);
  MatSetSizes(A, PETSC_DECIDE, PETSC_DECI
  MatSetFromOptions(A);
  MatSetUp(A);


  MatAssemblyBegin(); MatAssemblyEnd();


  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);
```

The i, j, and a arrays ARE copied by this routine into the internal format used by PETSc; thus you CAN matrix entries by changing the values of a[] after you have called this routine. Use MatCreateMPIAIJW avoid needing to copy the arrays.

The i and j indices are 0 based, and i indices are indices corresponding to the local j array.

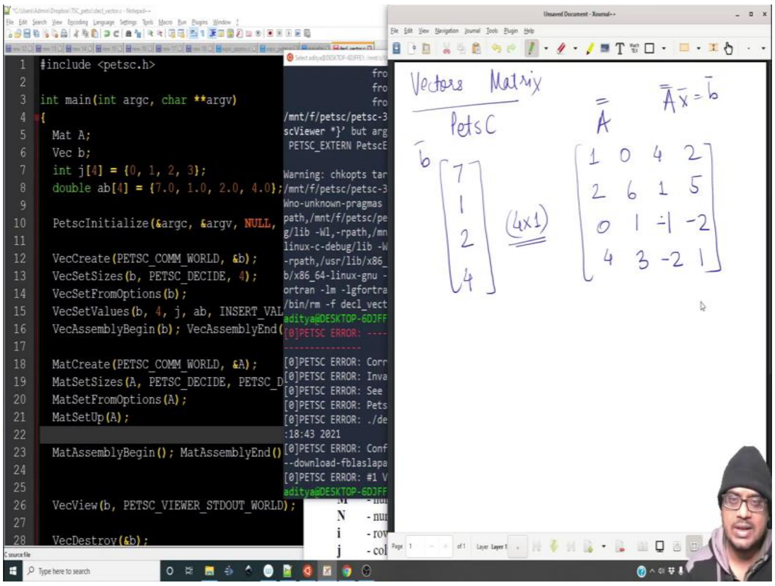
The format which is used for the sparse matrix input, is equivalent to a row-major ordering.. i.e for the input data expected is as shown

Once you have created the matrix you can update it with new numerical values using MatUpdateMPIA

```
   1 0 0

   2 0 3    P0

  -------

   4 5 6    P1

 Process0 [P0]: rows_owned=[0,1]

   i =  {0,1,3}  [size = nrow+1  = 2+1]

   j =  {0,0,2}  [size = 3]

   v =  {1,2,3}  [size = 3]

 Process1 [P1]: rows_owned=[2]

   i =  {0,3}    [size = nrow+1  = 1+1]
```

Let us so, in the standard CSR format you are going to create a MPIAIJ matrix. So, what it does is, it is going to; it is going to store what nonzero entries you have right in a single row. It is going to store the column number and the corresponding value; ok this is what it is alright.

(Refer Slide Time: 23:54)



```
#include <petsc.h>

int main(int argc, char **argv)
{
  Mat A;
  Vec b;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};

  PetscInitialize(&argc, &argv, NULL,

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VAL
  VecAssemblyBegin(b); VecAssemblyEnd(

  MatCreate(PETSC_COMM_WORLD, &A);
  MatSetSizes(A, PETSC_DECIDE, PETSC_D
  MatSetFromOptions(A);
  MatSetUp(A);


  MatAssemblyBegin(); MatAssemblyEnd()


  VecView(b, PETSC_VIEWER_STDOUT_WORLD);

  VecDestroy(&b);
```

```
            fro
            fro
            fro
/mnt/f/petsc/petsc-3
scViewer *}' but arg
 PETSC_EXTERN PetscE

Warning: chkopts tar
/mnt/f/petsc/petsc-3
Wno-unknown-pragmas
path,/mnt/f/petsc/pe
g/lib -Wl,-rpath,/mn
linux-c-debug/lib -W
-rpath,/usr/lib/x86_
b/x86_64-linux-gnu -
ortran -lm -lgfortra
/bin/rm -f decl_vect
aditya@DESKTOP-6DJFF
[0]PETSC ERROR: ----
---------------
[0]PETSC ERROR: Corr
[0]PETSC ERROR: Inva
[0]PETSC ERROR: See
[0]PETSC ERROR: Pets
[0]PETSC ERROR: ./de
:18:43 2021
[0]PETSC ERROR: Conf
--download-fblaslapa
[0]PETSC ERROR: #1 V
aditya@DESKTOP-6DJFF
```

N - nu
i - ro
j - col

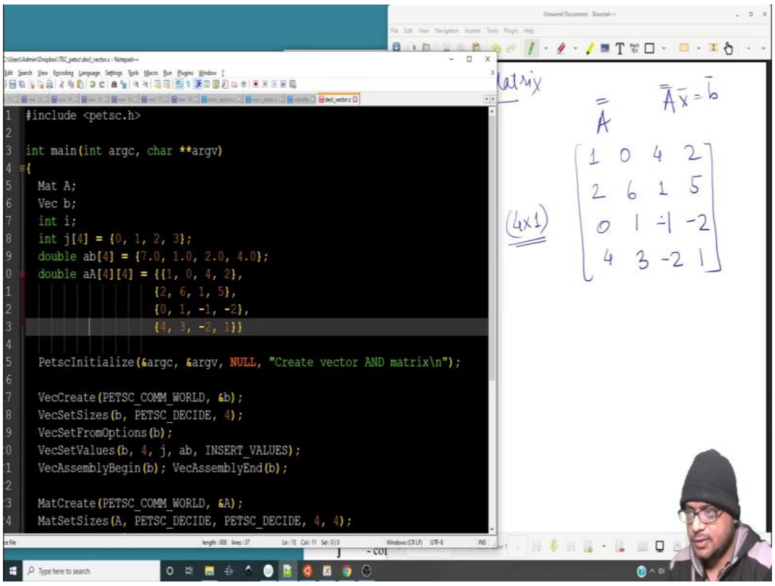
Vectors Matrix

PetsC

$$\bar{b} \begin{bmatrix} 7 \\ 1 \\ 2 \\ 4 \end{bmatrix} \quad (4\times1) \qquad \bar{\bar{A}} \begin{bmatrix} 1 & 0 & 4 & 2 \\ 2 & 6 & 1 & 5 \\ 0 & 1 & -1 & -2 \\ 4 & 3 & -2 & 1 \end{bmatrix} \qquad \bar{\bar{A}}\bar{x} = \bar{b}$$

So, so far we have not decided what the matrix should be. So, first we have to decide what the matrix has to be or let us create something 1 0 4 2 2 6 1 5 0 1 -1 -2 and 4 3 -2 1, I am hoping the determinant is an 0, because later we may be tempted to solve. So, if this is A this is b we will tempted to solve A x = b. Anyway, let us declare the matrix b like this. So, let me yeah got it, ok.

(Refer Slide Time: 24:54)



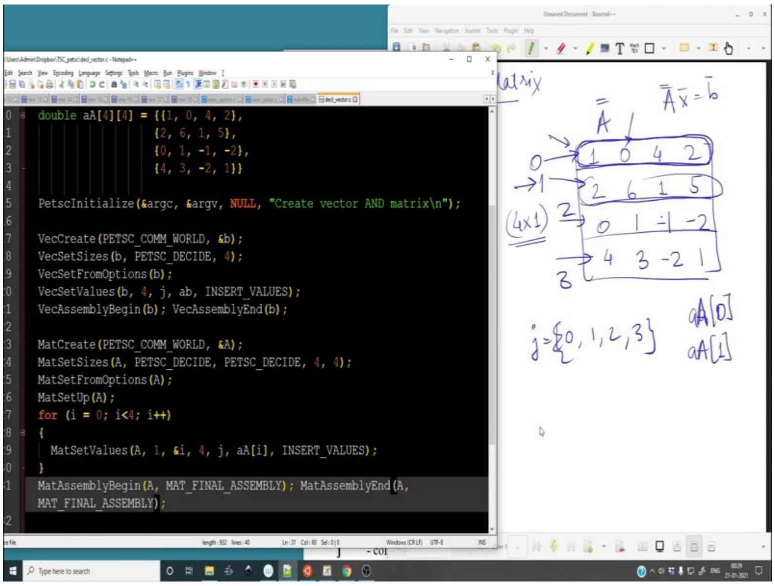
```c
#include <petsc.h>

int main(int argc, char **argv)
{
  Mat A;
  Vec b;
  int i;
  int j[4] = {0, 1, 2, 3};
  double ab[4] = {7.0, 1.0, 2.0, 4.0};
  double aA[4][4] = {{1, 0, 4, 2},
                     {2, 6, 1, 5},
                     {0, 1, -1, -2},
                     {4, 3, -2, 1}}

  PetscInitialize(&argc, &argv, NULL, "Create vector AND matrix\n");

  VecCreate(PETSC_COMM_WORLD, &b);
  VecSetSizes(b, PETSC_DECIDE, 4);
  VecSetFromOptions(b);
  VecSetValues(b, 4, j, ab, INSERT_VALUES);
  VecAssemblyBegin(b); VecAssemblyEnd(b);

  MatCreate(PETSC_COMM_WORLD, &A);
  MatSetSizes(A, PETSC_DECIDE, PETSC_DECIDE, 4, 4);
```

Matrix
$\bar{\bar{A}}$
$\bar{\bar{A}}\bar{x} = \bar{b}$
(4x1)
$\begin{bmatrix} 1 & 0 & 4 & 2 \\ 2 & 6 & 1 & 5 \\ 0 & 1 & -1 & -2 \\ 4 & 3 & -2 & 1 \end{bmatrix}$

So, now what we can do is create a double matrix. So, a A and size will be 4 × 4 and we can actually declare the entire array over here as well. So, what is this going to be 1 0 4 2 and we have 2 6 1 5 , and 0 1. Well, we are doing it the hard way, because it is so, small we can do it non-programmatically. Later on you will find tremendous benefit in doing it programmatically meaning; you run everything in a loop.

There is a distinct logic behind how these matrices will be set up, but in this case not no such thing exists. So, if the world so, we have defined the double array and yeah so, we need an integer to loop over 1 to 4 ok. So, we have an integer over here.

(Refer Slide Time: 26:26)



```
double aA[4][4] = {{1, 0, 4, 2},
                   {2, 6, 1, 5},
                   {0, 1, -1, -2},
                   {4, 3, -2, 1}}

PetscInitialize(&argc, &argv, NULL, "Create vector AND matrix\n");

VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUES);
VecAssemblyBegin(b); VecAssemblyEnd(b);

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DECIDE, 4, 4);
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i], INSERT_VALUES);
}
MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY); MatAssemblyEnd(A,
MAT_FINAL_ASSEMBLY);
```

So, what we can do over here in order to do this, so for i = 0 i < 4 i ++. Now, we can insert the elements into the matrix. So, the way to do it is MatSetValues into A. So, we are we doing it row by row the fact that I am using; the fact that I am using a loop I, is to loop over 1, 2 so, rather 0, 1, 2, 3. So, I am looping over 0, 1, 2, 3 through this particular for loop and in doing so, I am going to insert these particular values into this.

So, since we are doing it in a dense form, we are not really bothering about this 0 over here. It is a dense matrix. So, we are going to do j equal to 0 1 0, 1, 2, 3 which it already is. So, it already is 0, 1, 2, 3. So, we are going to take each row of this double array, a double pointer. So, we are going to take each row, insert it as column 0, 1, 2, 3, then take the first row insert 0, 1, 2, 3, second row insert 0, 1, 2, 3, third row insert 0, 1, 2, 3.

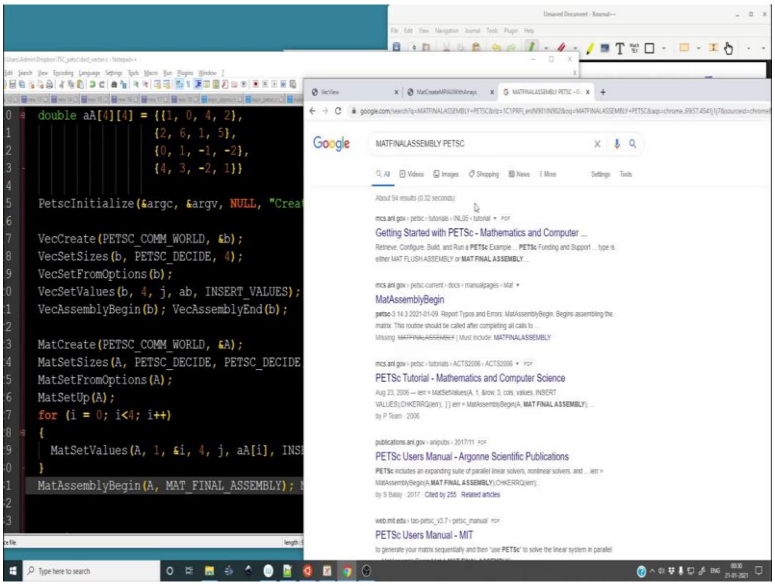
So, once I write it will be clear what I mean. So, I have, I am going to insert one row, I am going to pass the address of the row index that I want to insert. So, ampersand i means; the address of the current row index then I want to insert four values, I want to insert values into 0, 1, 2, 3. So, j is already an address. So, j so, thing is because i is an integer % i is the address, but because j is a j is an array rather j 4 is an array, j is the address, you do not need to pass an ampersand for the array alright.

So, these are some things which you pick up in C programming and lastly, we must pass the address of these fellows; this row, then this row, then this row, then this row, one by one and so, it is aA and it is simply going to be i and all is implied over here. So, in python you would have done this, but not over here that is implied. So, the address aA i holds these four elements. So, it is a double pointer so if the two square brackets, the two sets of square brackets means; there is a set of pointers which holds another set of point and which holds pointers to these rows ok.

So, meaning there is a pointer which holds these arrays and what holds this array? It is aA 1 rather aA 0; what holds this? aA 1 and so on. So, we are passing the address of those rows ok. So, this is how so, lastly we must tell insert values alright. So, yeah that is pretty much it with the help of this loop after. So, we have sort of set up what to insert and what where to insert.

So, now we must begin the matrix assembly. So, it is A ,MAT FINAL ASSEMBLY and you have to end also A, MAT FINAL ASSEMBLY alright. So, now you must be wondering why Final Assembly? Because there can be a sub matrix assembly as well.

(Refer Slide Time: 30:58)



```
double aA[4][4] = {{1, 0, 4, 2},
                   {2, 6, 1, 5},
                   {0, 1, -1, -2},
                   {4, 3, -2, 1}}

PetscInitialize(&argc, &argv, NULL, "Creat

VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUES);
VecAssemblyBegin(b); VecAssemblyEnd(b);

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DECIDE
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i], INS
}
MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY); M
```

MATFINALASSEMBLY PETSC

About 94 results (0.32 seconds)

mcs.anl.gov › petsc › tutorials › INL05 › tutorial PDF
Getting Started with PETSc - Mathematics and Computer ...
Retrieve, Configure, Build, and Run a **PETSc** Example ... **PETSc** Funding and Support ... type is either MAT FLUSH ASSEMBLY or **MAT FINAL ASSEMBLY** ...

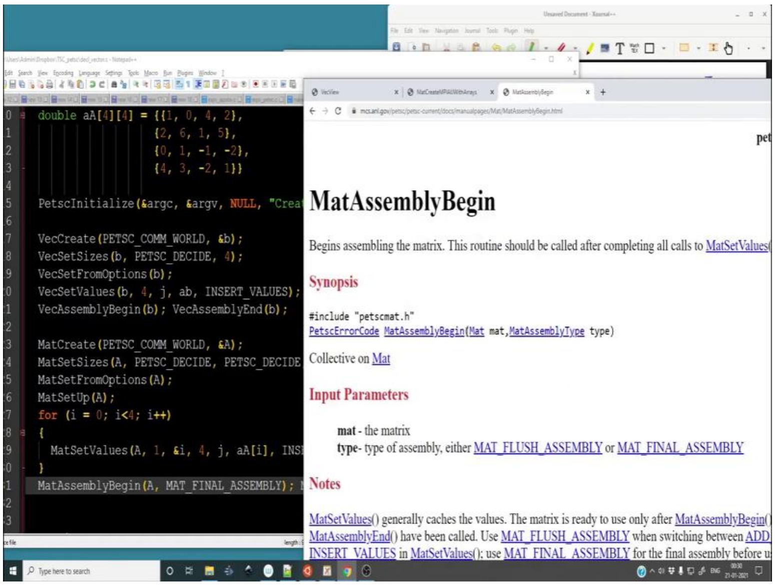
mcs.anl.gov › petsc-current › docs › manualpages › Mat
MatAssemblyBegin
petsc-3.14.3 2021-01-09. Report Typos and Errors. MatAssemblyBegin. Begins assembling the matrix. This routine should be called after completing all calls to ...
Missing: ~~MATFINALASSEMBLY~~ | Must include: MATFINALASSEMBLY

mcs.anl.gov › petsc › tutorials › ACTS2006 › ACTS2006 PDF
PETSc Tutorial - Mathematics and Computer Science
Aug 23, 2006 — ierr = MatSetValues(A, 1, &row, 3, cols, values, INSERT VALUES);CHKERRQ(ierr); } } ierr = MatAssemblyBegin(A, **MAT FINAL ASSEMBLY**); ...
by P Team · 2006

publications.anl.gov › anlpubs › 2017/11 PDF
PETSc Users Manual - Argonne Scientific Publications
PETSc includes an expanding suite of parallel linear solvers, nonlinear solvers, and ... ierr = MatAssemblyBegin(A,**MAT FINAL ASSEMBLY**);CHKERRQ(ierr);
by S Balay · 2017 · Cited by 255 · Related articles

web.mit.edu › tao-petsc_v3.7 › petsc_manual PDF
PETSc Users Manual - MIT
to generate your matrix sequentially and then "use **PETSc**" to solve the linear system in parallel.

(Refer Slide Time: 31:05)



```
double aA[4][4] = {{1, 0, 4, 2},
                   {2, 6, 1, 5},
                   {0, 1, -1, -2},
                   {4, 3, -2, 1}}

PetscInitialize(&argc, &argv, NULL, "Creat

VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUES);
VecAssemblyBegin(b); VecAssemblyEnd(b);

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DECIDE
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i], INS
}
MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY); M
```

MatAssemblyBegin

Begins assembling the matrix. This routine should be called after completing all calls to MatSetValues(

Synopsis

```
#include "petscmat.h"
PetscErrorCode MatAssemblyBegin(Mat mat,MatAssemblyType type)
```

Collective on Mat

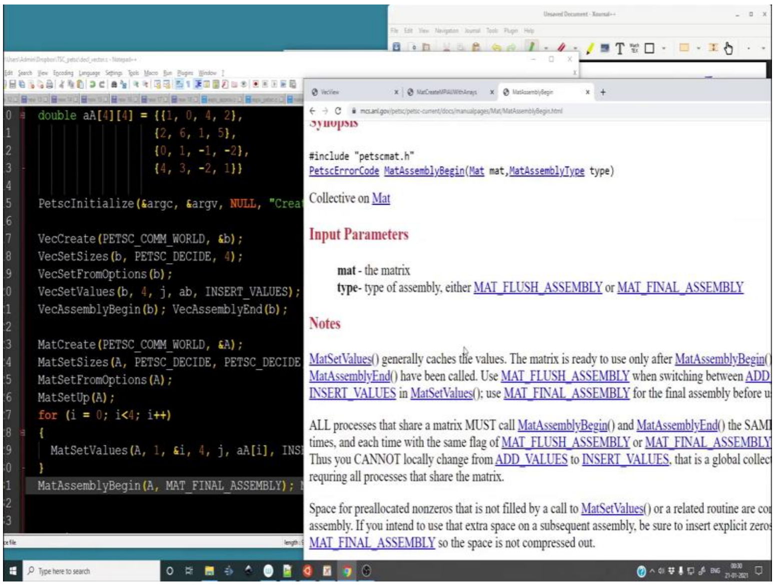
Input Parameters

mat - the matrix
type- type of assembly, either MAT_FLUSH_ASSEMBLY or MAT_FINAL_ASSEMBLY

Notes

MatSetValues() generally caches the values. The matrix is ready to use only after MatAssemblyBegin() MatAssemblyEnd() have been called. Use MAT_FLUSH_ASSEMBLY when switching between ADD INSERT_VALUES in MatSetValues(); use MAT_FINAL_ASSEMBLY for the final assembly before u

(Refer Slide Time: 31:06)



```
double aA[4][4] = {{1, 0, 4, 2},
                   {2, 6, 1, 5},
                   {0, 1, -1, -2},
                   {4, 3, -2, 1}}

PetscInitialize(&argc, &argv, NULL, "Creat

VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUES);
VecAssemblyBegin(b); VecAssemblyEnd(b);

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DECIDE
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i], INS
}
MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY); M
```

Synopsis

```
#include "petscmat.h"
PetscErrorCode MatAssemblyBegin(Mat mat,MatAssemblyType type)
```

Collective on Mat

Input Parameters

mat - the matrix
type- type of assembly, either MAT_FLUSH_ASSEMBLY or MAT_FINAL_ASSEMBLY

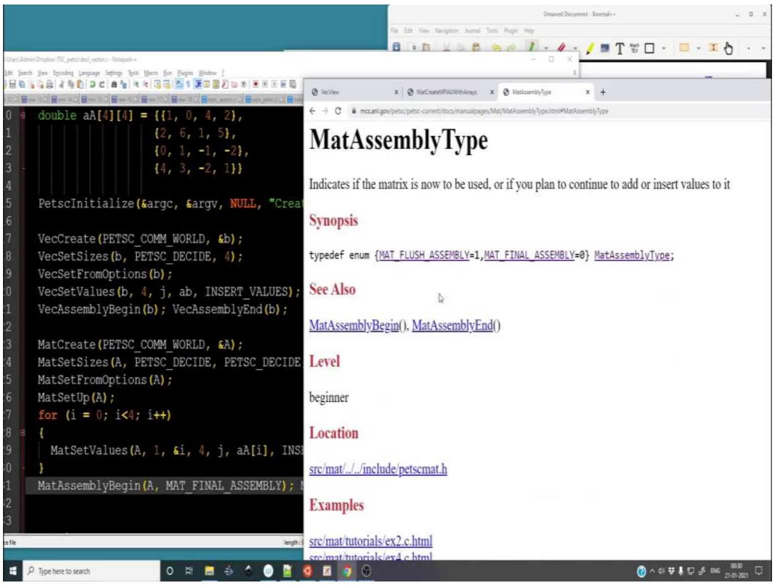
Notes

MatSetValues() generally caches the values. The matrix is ready to use only after MatAssemblyBegin() MatAssemblyEnd() have been called. Use MAT_FLUSH_ASSEMBLY when switching between ADD INSERT_VALUES in MatSetValues(); use MAT_FINAL_ASSEMBLY for the final assembly before u

ALL processes that share a matrix MUST call MatAssemblyBegin() and MatAssemblyEnd() the SAM times, and each time with the same flag of MAT_FLUSH_ASSEMBLY or MAT_FINAL_ASSEMBLY Thus you CANNOT locally change from ADD_VALUES to INSERT_VALUES, that is a global collect requring all processes that share the matrix.

Space for preallocated nonzeros that is not filled by a call to MatSetValues() or a related routine are co assembly. If you intend to use that extra space on a subsequent assembly, be sure to insert explicit zeros MAT_FINAL_ASSEMBLY so the space is not compressed out.

(Refer Slide Time: 31:10)



```
double aA[4][4] = {{1, 0, 4, 2},
                   {2, 6, 1, 5},
                   {0, 1, -1, -2},
                   {4, 3, -2, 1}}

PetscInitialize(&argc, &argv, NULL, "Creat

VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUES);
VecAssemblyBegin(b); VecAssemblyEnd(b);

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DECIDE
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i], INS
}
MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY); M
```

MatAssemblyType

Indicates if the matrix is now to be used, or if you plan to continue to add or insert values to it

Synopsis

`typedef enum {MAT_FLUSH_ASSEMBLY=1,MAT_FINAL_ASSEMBLY=0} MatAssemblyType;`

See Also

MatAssemblyBegin(), MatAssemblyEnd()

Level

beginner

Location

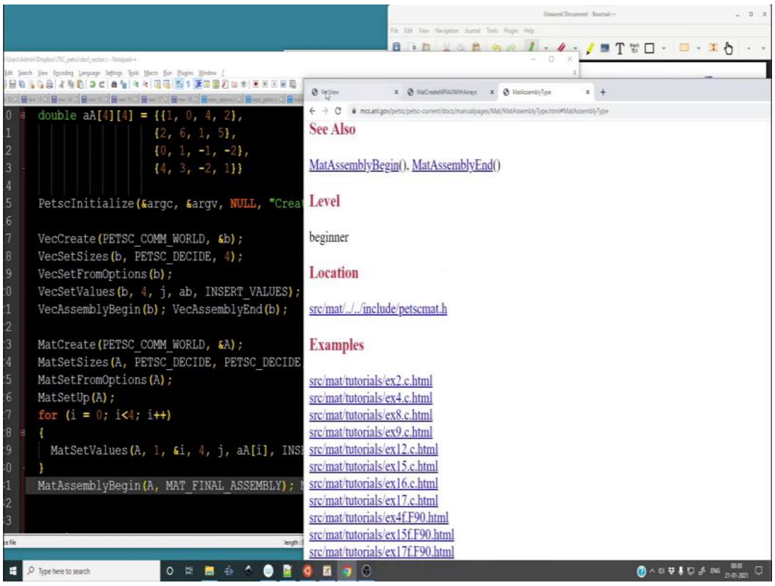
src/mat/../../include/petscmat.h

Examples

src/mat/tutorials/ex2.c.html
src/mat/tutorials/ex4.c.html

So, you so, it can be either flush assembly or final assembly ok, but in this particular examples, we just need to bother with final assembly alright.

(Refer Slide Time: 31:14)



```
double aA[4][4] = {{1, 0, 4, 2},
                   {2, 6, 1, 5},
                   {0, 1, -1, -2},
                   {4, 3, -2, 1}}

PetscInitialize(&argc, &argv, NULL, "Creat

VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUES);
VecAssemblyBegin(b); VecAssemblyEnd(b);

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DECIDE
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i], INS
}
MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY); M
```

See Also

MatAssemblyBegin(), MatAssemblyEnd()

Level

beginner

Location

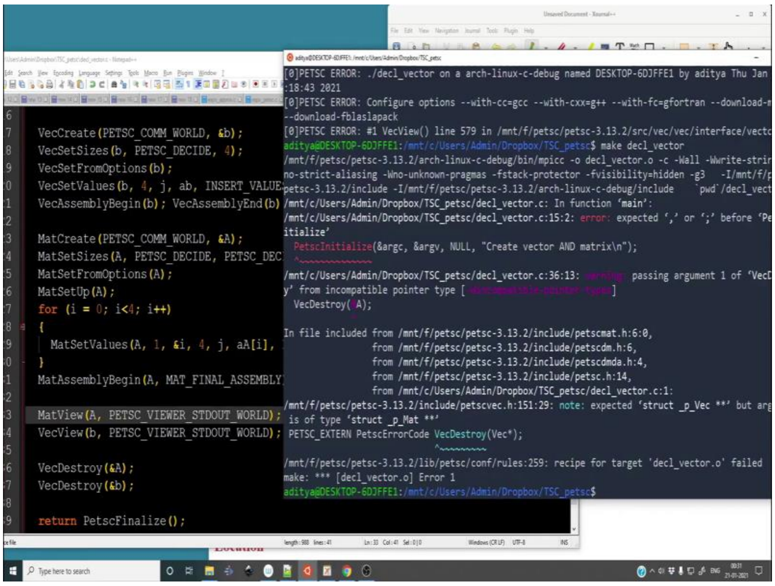
src/mat/../../include/petscmat.h

Examples

src/mat/tutorials/ex2.c.html
src/mat/tutorials/ex4.c.html
src/mat/tutorials/ex8.c.html
src/mat/tutorials/ex9.c.html
src/mat/tutorials/ex12.c.html
src/mat/tutorials/ex15.c.html
src/mat/tutorials/ex16.c.html
src/mat/tutorials/ex17.c.html
src/mat/tutorials/ex4f.F90.html
src/mat/tutorials/ex15f.F90.html
src/mat/tutorials/ex17f.F90.html

So, we can have a look at the function reference, we will figure it out. You can try various things, but yeah this is the general syntax for assembling a matrix.

(Refer Slide Time: 31:45)

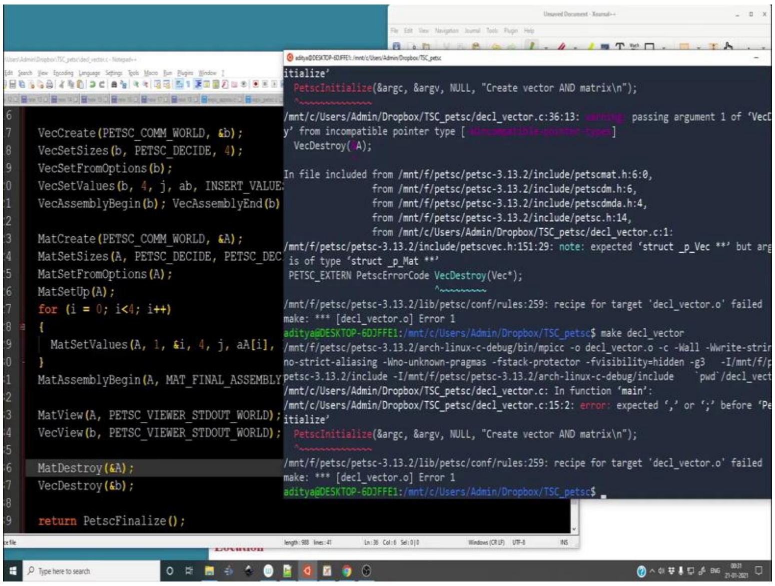


```
VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUE
VecAssemblyBegin(b); VecAssemblyEnd(b)

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DEC
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i],
}
MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY

MatView(A, PETSC_VIEWER_STDOUT_WORLD);
VecView(b, PETSC_VIEWER_STDOUT_WORLD);

VecDestroy(&A);
VecDestroy(&b);

return PetscFinalize();
```

```
[0]PETSC ERROR: ./decl_vector on a arch-linux-c-debug named DESKTOP-6DJFFE1 by aditya Thu Jan
:18:43 2021
[0]PETSC ERROR: Configure options --with-cc=gcc --with-cxx=g++ --with-fc=gfortran --download-m
--download-fblaslapack
[0]PETSC ERROR: #1 VecView() line 579 in /mnt/f/petsc/petsc-3.13.2/src/vec/vec/interface/vectc
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strin
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c: In function ‘main’:
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:15:2: error: expected ‘,’ or ‘;’ before ‘Pe
itialize’
  PetscInitialize(&argc, &argv, NULL, "Create vector AND matrix\n");
  ^~~~~~~~~~~~~~~
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:36:13: warning: passing argument 1 of ‘VecD
y’ from incompatible pointer type [-Wincompatible-pointer-types]
  VecDestroy(&A);

In file included from /mnt/f/petsc/petsc-3.13.2/include/petscmat.h:6:0,
                 from /mnt/f/petsc/petsc-3.13.2/include/petscdm.h:6,
                 from /mnt/f/petsc/petsc-3.13.2/include/petscdmda.h:4,
                 from /mnt/f/petsc/petsc-3.13.2/include/petsc.h:14,
                 from /mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:1:
/mnt/f/petsc/petsc-3.13.2/include/petscvec.h:151:29: note: expected ‘struct _p_Vec **’ but arg
 is of type ‘struct _p_Mat **’
 PETSC_EXTERN PetscErrorCode VecDestroy(Vec*);
                             ^~~~~~~~~~
/mnt/f/petsc/petsc-3.13.2/lib/petsc/conf/rules:259: recipe for target 'decl_vector.o' failed
make: *** [decl_vector.o] Error 1
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

So, once we have done this we must keep in mind that we must also destroy eventually, the pointer. So, the memory has to be cleared. So, we will simply pass the address of A and we can put a viewer over here as well. So, MatView A and we can use simply the STDOUT that is the terminal for visualizing the matrix ok.

(Refer Slide Time: 32:20)

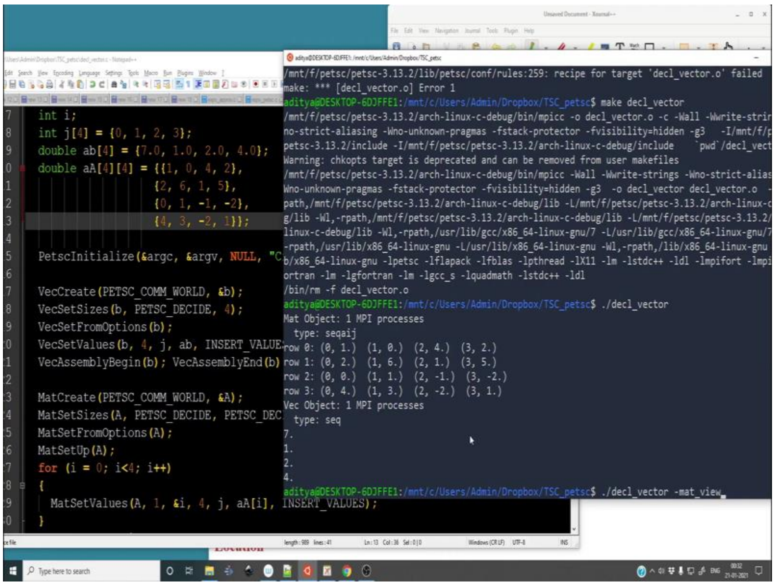


```
VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUE
VecAssemblyBegin(b); VecAssemblyEnd(b)

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DEC
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i],
}
MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY

MatView(A, PETSC_VIEWER_STDOUT_WORLD);
VecView(b, PETSC_VIEWER_STDOUT_WORLD);

MatDestroy(&A);
VecDestroy(&b);

return PetscFinalize();
```

```
itialize’
  PetscInitialize(&argc, &argv, NULL, "Create vector AND matrix\n");
  ^~~~~~~~~~~~~~~
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:36:13: warning: passing argument 1 of ‘VecD
y’ from incompatible pointer type [-Wincompatible-pointer-types]
  VecDestroy(&A);

In file included from /mnt/f/petsc/petsc-3.13.2/include/petscmat.h:6:0,
                 from /mnt/f/petsc/petsc-3.13.2/include/petscdm.h:6,
                 from /mnt/f/petsc/petsc-3.13.2/include/petscdmda.h:4,
                 from /mnt/f/petsc/petsc-3.13.2/include/petsc.h:14,
                 from /mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:1:
/mnt/f/petsc/petsc-3.13.2/include/petscvec.h:151:29: note: expected ‘struct _p_Vec **’ but arg
 is of type ‘struct _p_Mat **’
 PETSC_EXTERN PetscErrorCode VecDestroy(Vec*);
                             ^~~~~~~~~~
/mnt/f/petsc/petsc-3.13.2/lib/petsc/conf/rules:259: recipe for target 'decl_vector.o' failed
make: *** [decl_vector.o] Error 1
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strin
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c: In function ‘main’:
/mnt/c/Users/Admin/Dropbox/TSC_petsc/decl_vector.c:15:2: error: expected ‘,’ or ‘;’ before ‘Pe
itialize’
  PetscInitialize(&argc, &argv, NULL, "Create vector AND matrix\n");
  ^~~~~~~~~~~~~~~
/mnt/f/petsc/petsc-3.13.2/lib/petsc/conf/rules:259: recipe for target 'decl_vector.o' failed
make: *** [decl_vector.o] Error 1
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ _
```

There appears to be an there is it is not like destroy, but rather it has to be MatDestroy. There is a data type error ok. Let us make it ok, there is an error. Let us see what the error is. Have I missed a semicolon somewhere? We missed a semicolon over here ok.

(Refer Slide Time: 32:45)

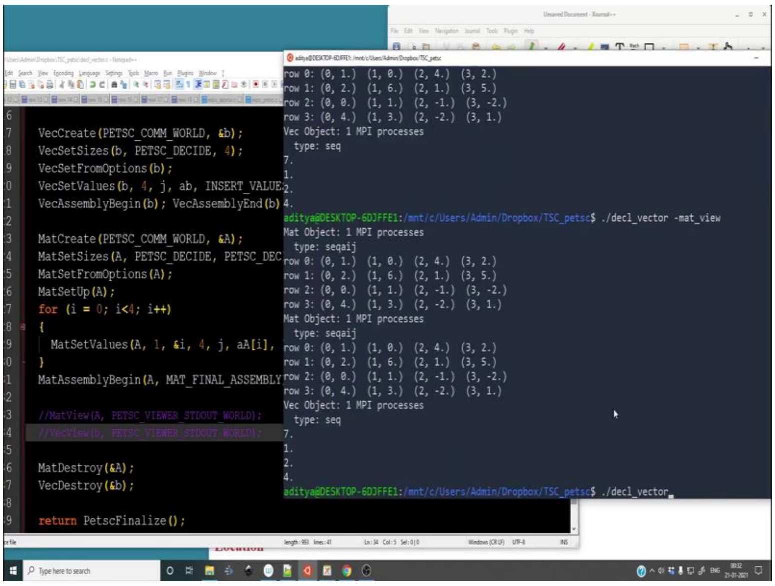


```
int i;
int j[4] = {0, 1, 2, 3};
double ab[4] = {7.0, 1.0, 2.0, 4.0};
double aA[4][4] = {{1, 0, 4, 2},
                   {2, 6, 1, 5},
                   {0, 1, -1, -2},
                   {4, 3, -2, 1}};

PetscInitialize(&argc, &argv, NULL, "C

VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUE
VecAssemblyBegin(b); VecAssemblyEnd(b)

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DEC
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i], INSERT_VALUES);
}
```

```
/mnt/f/petsc/petsc-3.13.2/lib/petsc/conf/rules:259: recipe for target 'decl_vector.o' failed
make: *** [decl_vector.o] Error 1
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strir
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
Warning: chkopts target is deprecated and can be removed from user makefiles
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
Mat Object: 1 MPI processes
  type: seqaij
row 0: (0, 1.)  (1, 0.)  (2, 4.)  (3, 2.)
row 1: (0, 2.)  (1, 6.)  (2, 1.)  (3, 5.)
row 2: (0, 0.)  (1, 1.)  (2, -1.)  (3, -2.)
row 3: (0, 4.)  (1, 3.)  (2, -2.)  (3, 1.)
Vec Object: 1 MPI processes
  type: seq
7.
1.
2.
4.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -mat_view
```

It is unforgiving in that sense, but yeah it is incredibly flexible ok. So, there you go so in row 0 the first row contains 1. So, it is CSR format so in row zeroth zeroth column has 1, first column has 0, second column has 4, third column has 2. So, the indexing; obviously, starts with 0 and similarly, you have all the other values. So, we can verify this. Now, what we can do is; we can print it over here as well.

(Refer Slide Time: 33:20)

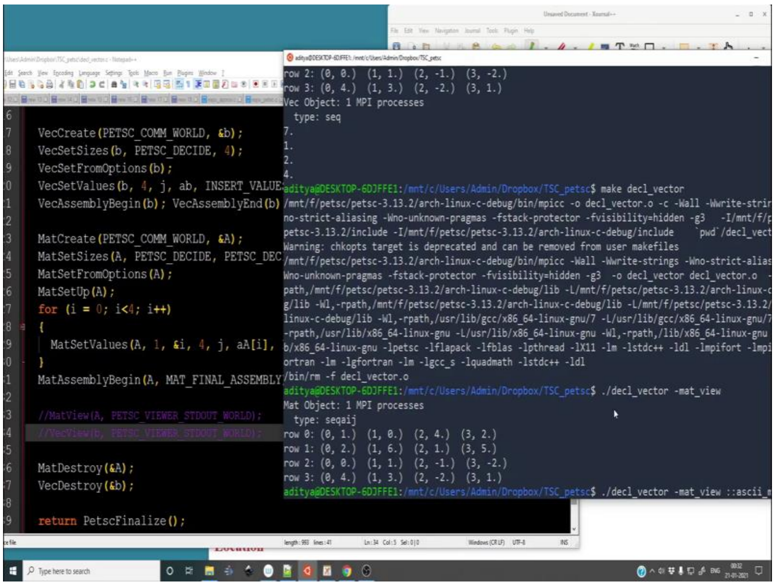


```

VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUE
VecAssemblyBegin(b); VecAssemblyEnd(b)

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DEC
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i],
}
MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY

//MatView(A, PETSC_VIEWER_STDOUT_WORLD);
//VecView(b, PETSC_VIEWER_STDOUT_WORLD);

MatDestroy(&A);
VecDestroy(&b);

return PetscFinalize();
```

```
row 0: (0, 1.)  (1, 0.)  (2, 4.)  (3, 2.)
row 1: (0, 2.)  (1, 6.)  (2, 1.)  (3, 5.)
row 2: (0, 0.)  (1, 1.)  (2, -1.)  (3, -2.)
row 3: (0, 4.)  (1, 3.)  (2, -2.)  (3, 1.)
Vec Object: 1 MPI processes
  type: seq
7.
1.
2.
4.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -mat_view
Mat Object: 1 MPI processes
  type: seqaij
row 0: (0, 1.)  (1, 0.)  (2, 4.)  (3, 2.)
row 1: (0, 2.)  (1, 6.)  (2, 1.)  (3, 5.)
row 2: (0, 0.)  (1, 1.)  (2, -1.)  (3, -2.)
row 3: (0, 4.)  (1, 3.)  (2, -2.)  (3, 1.)
Mat Object: 1 MPI processes
  type: seqaij
row 0: (0, 1.)  (1, 0.)  (2, 4.)  (3, 2.)
row 1: (0, 2.)  (1, 6.)  (2, 1.)  (3, 5.)
row 2: (0, 0.)  (1, 1.)  (2, -1.)  (3, -2.)
row 3: (0, 4.)  (1, 3.)  (2, -2.)  (3, 1.)
Vec Object: 1 MPI processes
  type: seq
7.
1.
2.
4.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector
```

So, you can; so you can comment out these lines and print out by passing the command line argument.

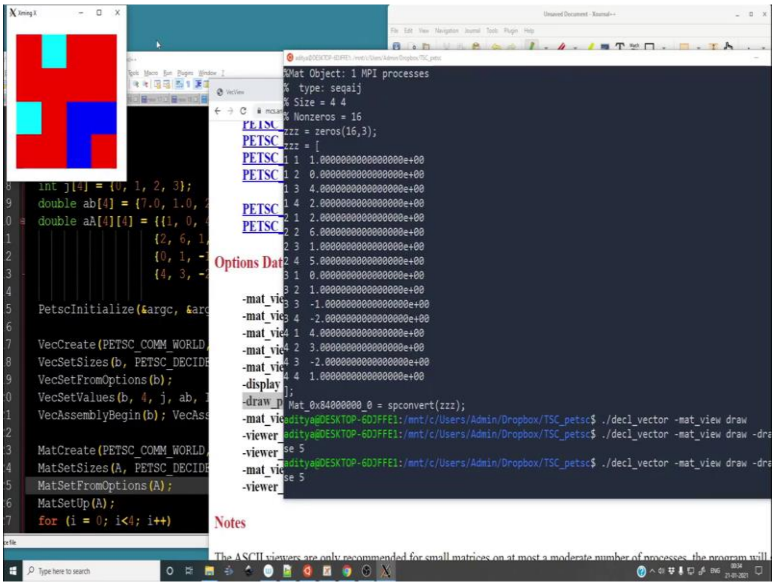
(Refer Slide Time: 33:31)



```
VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUE
VecAssemblyBegin(b); VecAssemblyEnd(b)

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DEC
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i],
}
MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY

//MatView(A, PETSC_VIEWER_STDOUT_WORLD);
//VecView(b, PETSC_VIEWER_STDOUT_WORLD);

MatDestroy(&A);
VecDestroy(&b);

return PetscFinalize();
```

```
row 2: (0, 0.)  (1, 1.)  (2, -1.)  (3, -2.)
row 3: (0, 4.)  (1, 3.)  (2, -2.)  (3, 1.)
Vec Object: 1 MPI processes
  type: seq
7.
1.
2.
4.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make decl_vector
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o decl_vector.o -c -Wall -Wwrite-strir
no-strict-aliasing -Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/p
petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/include    `pwd`/decl_vect
Warning: chkopts target is deprecated and can be removed from user makefiles
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -Wall -Wwrite-strings -Wno-strict-alias
Wno-unknown-pragmas -fstack-protector -fvisibility=hidden -g3  -o decl_vector decl_vector.o  -
path,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c
g/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt/f/petsc/petsc-3.13.2/
linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7
-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu
b/x86_64-linux-gnu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi
ortran -lm -lgfortran -lm -lgcc_s -lquadmath -lstdc++ -ldl
/bin/rm -f decl_vector.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -mat_view
Mat Object: 1 MPI processes
  type: seqaij
row 0: (0, 1.)  (1, 0.)  (2, 4.)  (3, 2.)
row 1: (0, 2.)  (1, 6.)  (2, 1.)  (3, 5.)
row 2: (0, 0.)  (1, 1.)  (2, -1.)  (3, -2.)
row 3: (0, 4.)  (1, 3.)  (2, -2.)  (3, 1.)
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -mat_view ::ascii_
```

(Refer Slide Time: 33:44)

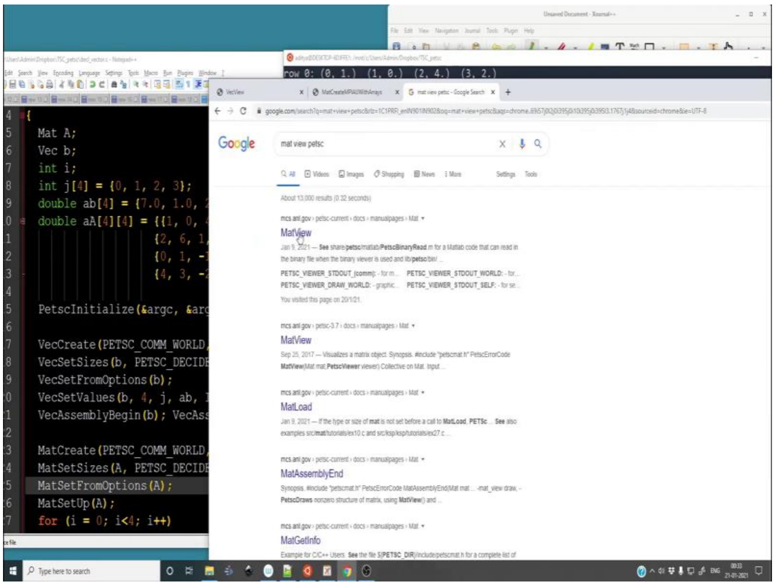


```
int j[4] = {0, 1, 2, 3};
double ab[4] = {7.0, 1.0, 
double aA[4][4] = {{1, 0, 
                   {2, 6, 1,
                   {0, 1, -
                   {4, 3, -2

PetscInitialize(&argc, &arg

VecCreate(PETSC_COMM_WORLD,
VecSetSizes(b, PETSC_DECIDE
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, I
VecAssemblyBegin(b); VecAss

MatCreate(PETSC_COMM_WORLD,
MatSetSizes(A, PETSC_DECIDE
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
```

```
%Mat Object: 1 MPI processes
%  type: seqaij
% Size = 4 4
% Nonzeros = 16
zzz = zeros(16,3);
zzz = [
1 1  1.0000000000000000e+00
1 2  0.0000000000000000e+00
1 3  4.0000000000000000e+00
1 4  2.0000000000000000e+00
2 1  2.0000000000000000e+00
2 2  6.0000000000000000e+00
2 3  1.0000000000000000e+00
2 4  5.0000000000000000e+00
3 1  0.0000000000000000e+00
3 2  1.0000000000000000e+00
3 3  -1.0000000000000000e+00
3 4  -2.0000000000000000e+00
4 1  4.0000000000000000e+00
4 2  3.0000000000000000e+00
4 3  -2.0000000000000000e+00
4 4  1.0000000000000000e+00
];
 Mat_0x84000000_0 = spconvert(zzz);
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -mat_view draw
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -mat_view draw -dra
se 5
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -mat_view draw -dra
se 5
```

So, let us save this, let us recompile and let us call it with this. So, you can simply print the MatView, we can print it as ascii matlab this is the MATLAB way of writing it ok. So, MATLAB indexing starts from 1 1 , 1 is value 1. So, let us see 1 , 1 value is 1, 1 0 4 2 2 6 1 5 and so on. So, this is how MATLAB would represent it. There is also a way of visualizing this entire thing.

(Refer Slide Time: 34:11)



```
{
    Mat A;
    Vec b;
    int i;
    int j[4] = {0, 1, 2, 3};
    double ab[4] = {7.0, 1.0, 2
    double aA[4][4] = {{1, 0, 4
                        {2, 6, 1,
                        {0, 1, -1
                        {4, 3, -2

    PetscInitialize(&argc, &arg

    VecCreate(PETSC_COMM_WORLD,
    VecSetSizes(b, PETSC_DECIDE
    VecSetFromOptions(b);
    VecSetValues(b, 4, j, ab, I
    VecAssemblyBegin(b); VecAss

    MatCreate(PETSC_COMM_WORLD,
    MatSetSizes(A, PETSC_DECIDE
    MatSetFromOptions(A);
    MatSetUp(A);
    for (i = 0; i<4; i++)
```

```
row 0: (0, 1.)  (1, 0.)  (2, 4.)  (3, 2.)
```

google.com/search?q=mat+view+petsc

mat view petsc

About 13,000 results (0.32 seconds)

mcs.anl.gov › petsc-current › docs › manualpages › Mat
MatView
Jan 9, 2021 — See share/petsc/matlab/PetscBinaryRead.m for a Matlab code that can read in the binary file when the binary viewer is used and lib/petsc/bin/ ...
PETSC_VIEWER_STDOUT_(comm): - for m... PETSC_VIEWER_STDOUT_WORLD: - for...
PETSC_VIEWER_DRAW_WORLD: - graphic... PETSC_VIEWER_STDOUT_SELF: - for se...
You visited this page on 20/1/21.

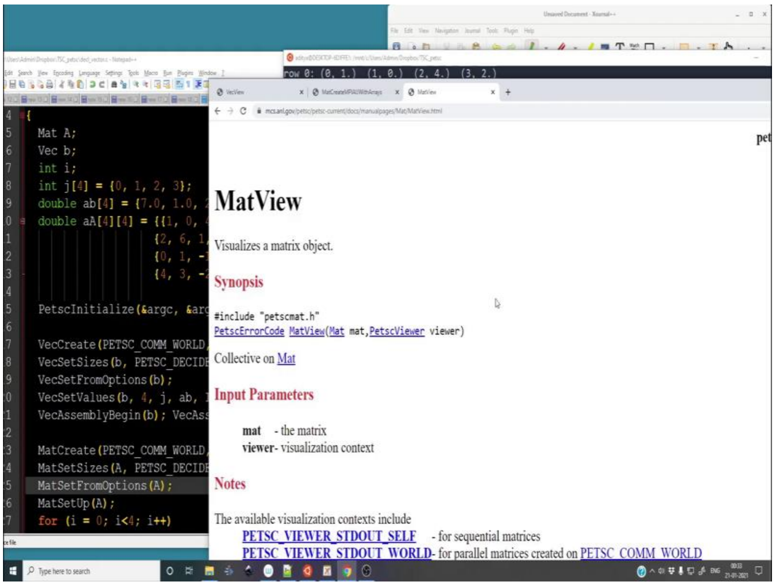
mcs.anl.gov › petsc-3.7 › docs › manualpages › Mat
MatView
Sep 25, 2017 — Visualizes a matrix object. Synopsis. #include "petscmat.h" PetscErrorCode MatView(Mat mat,PetscViewer viewer) Collective on Mat. Input ...

mcs.anl.gov › petsc-current › docs › manualpages › Mat
MatLoad
Jan 9, 2021 — If the type or size of mat is not set before a call to MatLoad, PETSc ... See also examples src/mat/tutorials/ex10.c and src/ksp/ksp/tutorials/ex27.c ...

mcs.anl.gov › petsc-current › docs › manualpages › Mat
MatAssemblyEnd
Synopsis. #include "petscmat.h" PetscErrorCode MatAssemblyEnd(Mat mat ... -mat_view draw, - PetscDraws nonzero structure of matrix, using MatView() and ...

mcs.anl.gov › petsc-current › docs › manualpages › Mat
MatGetInfo
Example for C/C++ Users. See the file ${PETSC_DIR}/include/petscmat.h for a complete list of

(Refer Slide Time: 34:17)



```
{
    Mat A;
    Vec b;
    int i;
    int j[4] = {0, 1, 2, 3};
    double ab[4] = {7.0, 1.0, 2
    double aA[4][4] = {{1, 0, 4
                        {2, 6, 1,
                        {0, 1, -1
                        {4, 3, -2

    PetscInitialize(&argc, &arg

    VecCreate(PETSC_COMM_WORLD,
    VecSetSizes(b, PETSC_DECIDE
    VecSetFromOptions(b);
    VecSetValues(b, 4, j, ab, I
    VecAssemblyBegin(b); VecAss

    MatCreate(PETSC_COMM_WORLD,
    MatSetSizes(A, PETSC_DECIDE
    MatSetFromOptions(A);
    MatSetUp(A);
    for (i = 0; i<4; i++)
```

```
row 0: (0, 1.)  (1, 0.)  (2, 4.)  (3, 2.)
```

mcs.anl.gov/petsc/petsc-current/docs/manualpages/Mat/MatView.html

MatView

Visualizes a matrix object.

Synopsis

```
#include "petscmat.h"
PetscErrorCode MatView(Mat mat,PetscViewer viewer)
```

Collective on Mat

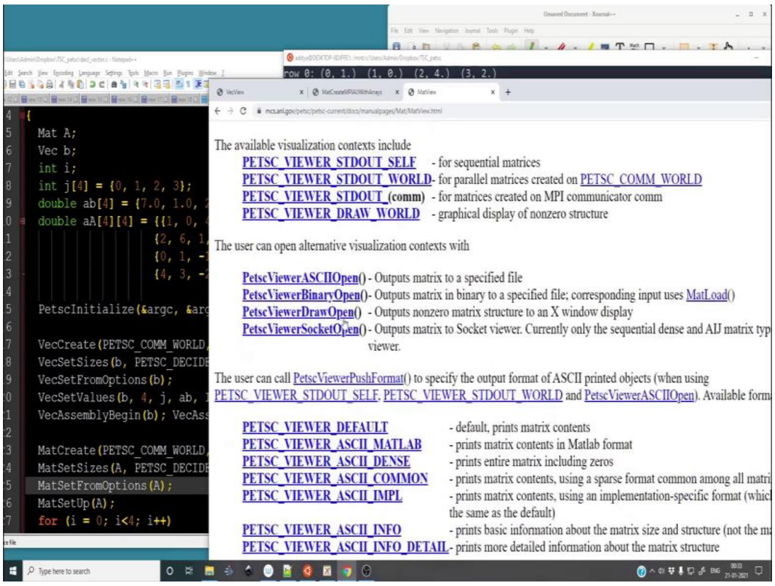
Input Parameters

mat - the matrix
viewer- visualization context

Notes

The available visualization contexts include
PETSC_VIEWER_STDOUT_SELF - for sequential matrices
PETSC_VIEWER_STDOUT_WORLD- for parallel matrices created on PETSC_COMM_WORLD

(Refer Slide Time: 34:18)



```
row 0: (0, 1.)  (1, 0.)  (2, 4.)  (3, 2.)
```

mcs.anl.gov/petsc/petsc-current/docs/manualpages/Mat/MatView.html

```
{
    Mat A;
    Vec b;
    int i;
    int j[4] = {0, 1, 2, 3};
    double ab[4] = {7.0, 1.0, 2
    double aA[4][4] = {{1, 0, 4
                        {2, 6, 1
                        {0, 1, -1
                        {4, 3, -2

    PetscInitialize(&argc, &arg

    VecCreate(PETSC_COMM_WORLD,
    VecSetSizes(b, PETSC_DECIDE
    VecSetFromOptions(b);
    VecSetValues(b, 4, j, ab, I
    VecAssemblyBegin(b); VecAss

    MatCreate(PETSC_COMM_WORLD,
    MatSetSizes(A, PETSC_DECIDE
    MatSetFromOptions(A);
    MatSetUp(A);
    for (i = 0; i<4; i++)
```

The available visualization contexts include

PETSC_VIEWER_STDOUT_SELF - for sequential matrices
PETSC_VIEWER_STDOUT_WORLD- for parallel matrices created on PETSC_COMM_WORLD
PETSC_VIEWER_STDOUT_(comm) - for matrices created on MPI communicator comm
PETSC_VIEWER_DRAW_WORLD - graphical display of nonzero structure

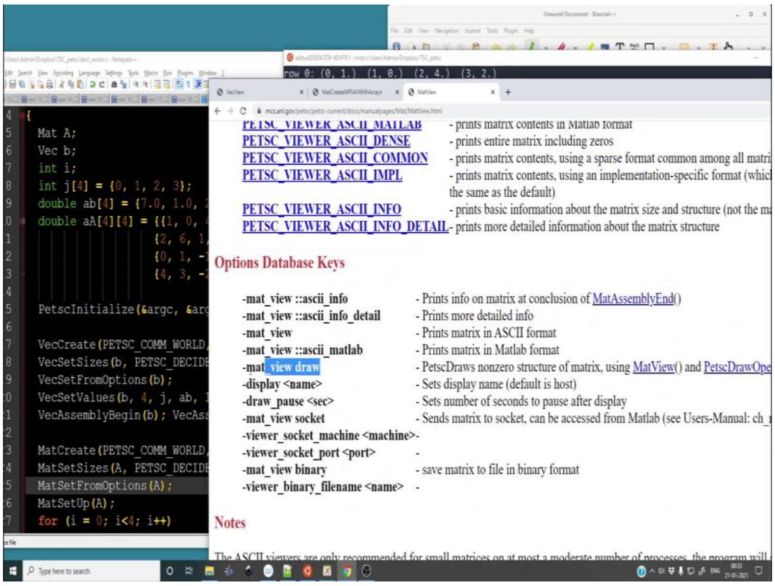
The user can open alternative visualization contexts with

PetscViewerASCIIOpen() - Outputs matrix to a specified file
PetscViewerBinaryOpen()- Outputs matrix in binary to a specified file; corresponding input uses MatLoad()
PetscViewerDrawOpen() - Outputs nonzero matrix structure to an X window display
PetscViewerSocketOpen() - Outputs matrix to Socket viewer. Currently only the sequential dense and AIJ matrix typ viewer.

The user can call PetscViewerPushFormat() to specify the output format of ASCII printed objects (when using PETSC_VIEWER_STDOUT_SELF, PETSC_VIEWER_STDOUT_WORLD and PetscViewerASCIIOpen). Available form

PETSC_VIEWER_DEFAULT - default, prints matrix contents
PETSC_VIEWER_ASCII_MATLAB - prints matrix contents in Matlab format
PETSC_VIEWER_ASCII_DENSE - prints entire matrix including zeros
PETSC_VIEWER_ASCII_COMMON - prints matrix contents, using a sparse format common among all matri
PETSC_VIEWER_ASCII_IMPL - prints matrix contents, using an implementation-specific format (whic the same as the default)
PETSC_VIEWER_ASCII_INFO - prints basic information about the matrix size and structure (not the m
PETSC_VIEWER_ASCII_INFO_DETAIL- prints more detailed information about the matrix structure

(Refer Slide Time: 34:30)



```
row 0: (0, 1.)  (1, 0.)  (2, 4.)  (3, 2.)
```

mcs.anl.gov/petsc/petsc-current/docs/manualpages/Mat/MatView.html

```
{
    Mat A;
    Vec b;
    int i;
    int j[4] = {0, 1, 2, 3};
    double ab[4] = {7.0, 1.0, 2
    double aA[4][4] = {{1, 0, 4
                        {2, 6, 1
                        {0, 1, -1
                        {4, 3, -2

    PetscInitialize(&argc, &arg

    VecCreate(PETSC_COMM_WORLD,
    VecSetSizes(b, PETSC_DECIDE
    VecSetFromOptions(b);
    VecSetValues(b, 4, j, ab, I
    VecAssemblyBegin(b); VecAss

    MatCreate(PETSC_COMM_WORLD,
    MatSetSizes(A, PETSC_DECIDE
    MatSetFromOptions(A);
    MatSetUp(A);
    for (i = 0; i<4; i++)
```

PETSC_VIEWER_ASCII_MATLAB - prints matrix contents in Matlab format
PETSC_VIEWER_ASCII_DENSE - prints entire matrix including zeros
PETSC_VIEWER_ASCII_COMMON - prints matrix contents, using a sparse format common among all matri
PETSC_VIEWER_ASCII_IMPL - prints matrix contents, using an implementation-specific format (whic the same as the default)
PETSC_VIEWER_ASCII_INFO - prints basic information about the matrix size and structure (not the m
PETSC_VIEWER_ASCII_INFO_DETAIL- prints more detailed information about the matrix structure

Options Database Keys

-mat_view ::ascii_info - Prints info on matrix at conclusion of MatAssemblyEnd()
-mat_view ::ascii_info_detail - Prints more detailed info
-mat_view - Prints matrix in ASCII format
-mat_view ::ascii_matlab - Prints matrix in Matlab format
-mat_view draw - PetscDraws nonzero structure of matrix, using MatView() and PetscDrawOpe
-display <name> - Sets display name (default is host)
-draw_pause <sec> - Sets number of seconds to pause after display
-mat_view socket - Sends matrix to socket, can be accessed from Matlab (see Users-Manual: ch_
-viewer_socket_machine <machine>-
-viewer_socket_port <port> -
-mat_view binary - save matrix to file in binary format
-viewer_binary_filename <name> -

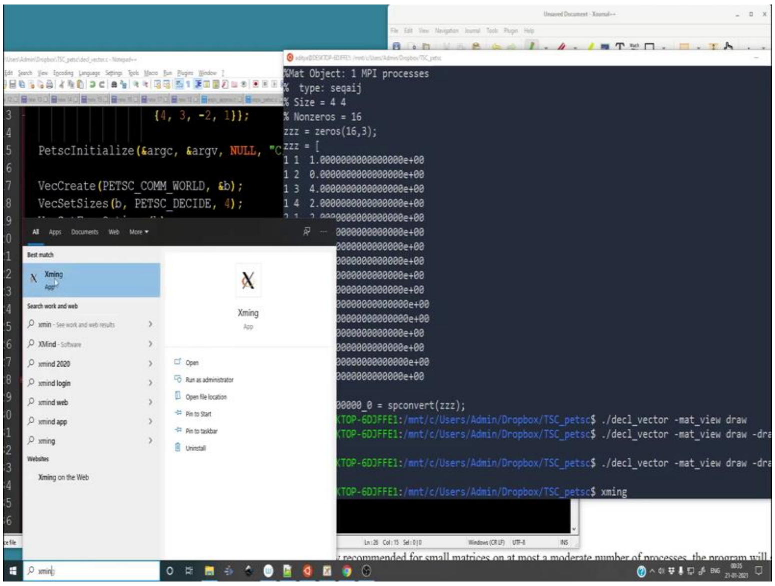
Notes

The ASCII viewers are only recommended for small matrices on at most a moderate number of processes, the program will

So, let me go to the function reference for mat view ok. So, what do we have? Let us see we will draw a world ok, but we can we can do this mat view draw. So, what we can do is mat view draw, quickly we came and went away. So, we must have a draw pause as well. So, look there is a draw pause as well which we need to do. So, draw pause of 5 seconds ok. So, this is how it looks.

So, the 0 is cyan, positive values are red, the negative values are blue. I will do it again in case you want to see, great. So, we have seen how to create a vector, how to create a matrix, how to assemble it. There are many ways of doing it and in the coming lectures you will see some other ways of doing it as well. In general you will see a programmatic way of doing it. One last thing I want to point out the graphics that you just saw, for that you need to install some kind of X 11 forwarder.

(Refer Slide Time: 35:41)

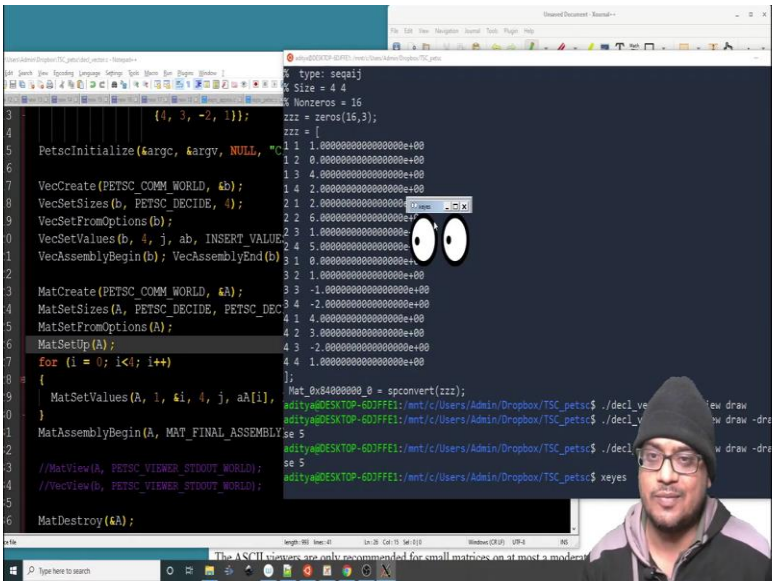


```
                    {4, 3, -2, 1}};

    PetscInitialize(&argc, &argv, NULL, "C

    VecCreate(PETSC_COMM_WORLD, &b);
    VecSetSizes(b, PETSC_DECIDE, 4);
```

```
%Mat Object: 1 MPI processes
%  type: seqaij
% Size = 4 4
% Nonzeros = 16
zzz = zeros(16,3);
zzz = [
1 1  1.0000000000000000e+00
1 2  0.0000000000000000e+00
1 3  4.0000000000000000e+00
1 4  2.0000000000000000e+00
2 1  2.0000000000000000e+00
     0000000000000e+00
     0000000000000e+00
     0000000000000e+00
     0000000000000e+00
     0000000000000e+00
     0000000000000e+00
     0000000000000e+00
     0000000000000e+00
     0000000000000e+00
     0000000000000e+00
     0000000000000e+00

00000_0 = spconvert(zzz);
KTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -mat_view draw
KTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -mat_view draw -dra

KTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -mat_view draw -dra

KTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ xming
```

So, because we are using sort of a virtualization of Linux, you need to install something like Xming or something whatever you like you have to open it up. So, Xming Xming I have it installed. So, you have to open this, then launch the or you can launch ubuntu, then launch Xming and how to test whether Xming is installed you just say xeyes.

(Refer Slide Time: 36:08)



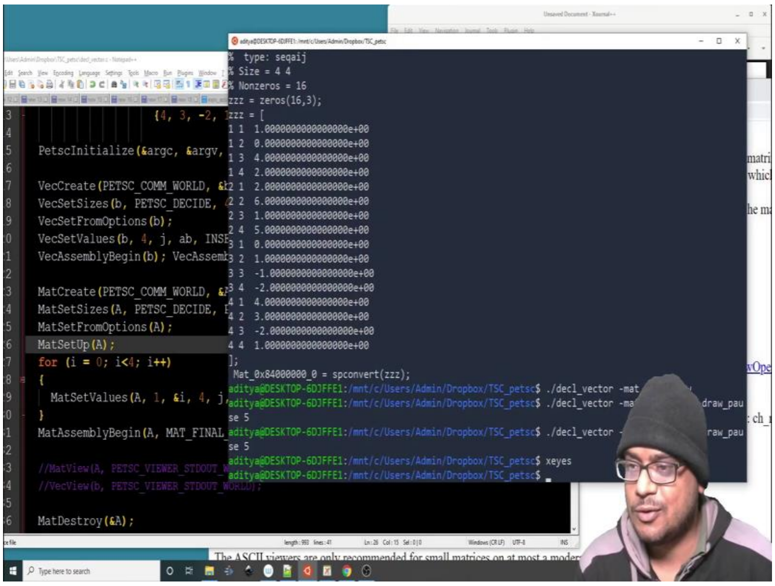
```
                        {4, 3, -2, 1}};

PetscInitialize(&argc, &argv, NULL, "C

VecCreate(PETSC_COMM_WORLD, &b);
VecSetSizes(b, PETSC_DECIDE, 4);
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INSERT_VALUE
VecAssemblyBegin(b); VecAssemblyEnd(b)

MatCreate(PETSC_COMM_WORLD, &A);
MatSetSizes(A, PETSC_DECIDE, PETSC_DEC
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j, aA[i],
}
MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY

//MatView(A, PETSC_VIEWER_STDOUT_WORLD);
//VecView(b, PETSC_VIEWER_STDOUT_WORLD);

MatDestroy(&A);
```

```
% type: seqaij
% Size = 4 4
% Nonzeros = 16
zzz = zeros(16,3);
zzz = [
1 1  1.0000000000000000e+00
1 2  0.0000000000000000e+00
1 3  4.0000000000000000e+00
1 4  2.0000000000000000e+00
2 1  2.0000000000000000e
2 2  6.0000000000000000e+
2 3  1.0000000000000000e
2 4  5.0000000000000000e
3 1  0.0000000000000000e+
3 2  1.0000000000000000e+00
3 3  -1.0000000000000000e+00
3 4  -2.0000000000000000e+00
4 1  4.0000000000000000e+00
4 2  3.0000000000000000e+00
4 3  -2.0000000000000000e+00
4 4  1.0000000000000000e+00
];
 Mat_0x84000000_0 = spconvert(zzz);
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_ve          iew draw
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_v          ew draw -dra
se 5
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl           w draw -dra
se 5
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ xeyes
```

The ASCII viewers are only recommended for small matrices on at most a moderat

And you will have this funny X 11 application in Linux, if this works everything is fine, if this does not work then you will not be able to visualize things natively from this. But for those of you who are working natively in Linux all these things would not matter. And I am too lazy to install the whole thing the petsc library is natively in windows. So, that is why I went for virtualization.

(Refer Slide Time: 36:18)



```
                        {4, 3, -2,

PetscInitialize(&argc, &argv,

VecCreate(PETSC_COMM_WORLD, &
VecSetSizes(b, PETSC_DECIDE,
VecSetFromOptions(b);
VecSetValues(b, 4, j, ab, INS
VecAssemblyBegin(b); VecAssem

MatCreate(PETSC_COMM_WORLD, &
MatSetSizes(A, PETSC_DECIDE,
MatSetFromOptions(A);
MatSetUp(A);
for (i = 0; i<4; i++)
{
  MatSetValues(A, 1, &i, 4, j
}
MatAssemblyBegin(A, MAT_FINAL

//MatView(A, PETSC_VIEWER_STDOUT_
//VecView(b, PETSC_VIEWER_STDOUT_WORLD);

MatDestroy(&A);
```

```
% type: seqaij
% Size = 4 4
% Nonzeros = 16
zzz = zeros(16,3);
zzz = [
1 1  1.0000000000000000e+00
1 2  0.0000000000000000e+00
1 3  4.0000000000000000e+00
1 4  2.0000000000000000e+00
2 1  2.0000000000000000e+00
2 2  6.0000000000000000e+00
2 3  1.0000000000000000e+00
2 4  5.0000000000000000e+00
3 1  0.0000000000000000e+00
3 2  1.0000000000000000e+00
3 3  -1.0000000000000000e+00
3 4  -2.0000000000000000e+00
4 1  4.0000000000000000e+00
4 2  3.0000000000000000e+00
4 3  -2.0000000000000000e+00
4 4  1.0000000000000000e+00
];
 Mat_0x84000000_0 = spconvert(zzz);
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -mat
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -ma          draw_pau
se 5
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./decl_vector -          raw_pau
se 5
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ xeyes
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ _
```

The ASCII viewers are only recommended for small matrices on at most a moder

Anyway; so, I hope you will find whatever I have shown useful and practice makes you perfect. So, just go ahead and do various kinds of matrices that you like. In the next class, we will see how to make a simple solver and we will proceed with a 1D problem. With this, I will see you next time until then, good bye.