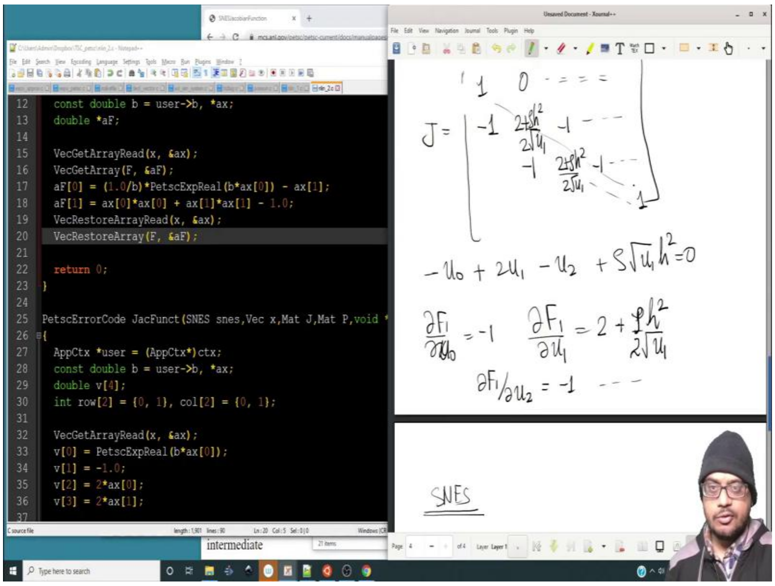
Tools in Scientific Computing
Prof. Aditya Bandopadhyay
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 35
Reaction-Diffusion system in PETSc

(Refer Slide Time: 00:30)



```
  const double b = user->b, *ax;
  double *aF;

  VecGetArrayRead(x, &ax);
  VecGetArray(F, &aF);
  aF[0] = (1.0/b)*PetscExpReal(b*ax[0]) - ax[1];
  aF[1] = ax[0]*ax[0] + ax[1]*ax[1] - 1.0;
  VecRestoreArrayRead(x, &ax);
  VecRestoreArray(F, &aF);

  return 0;
}

PetscErrorCode JacFunct(SNES snes,Vec x,Mat J,Mat P,void *
{
  AppCtx *user = (AppCtx*)ctx;
  const double b = user->b, *ax;
  double v[4];
  int row[2] = {0, 1}, col[2] = {0, 1};

  VecGetArrayRead(x, &ax);
  v[0] = PetscExpReal(b*ax[0]);
  v[1] = -1.0;
  v[2] = 2*ax[0];
  v[3] = 2*ax[1];
```

$$J = \begin{vmatrix} 1 & 0 & \cdots & & \\ -1 & 2+\frac{Sh^2}{2\sqrt{u_1}} & -1 & \cdots & \\ & -1 & 2+\frac{Sh^2}{2\sqrt{u_1}} & -1 & \cdots \\ & & & \ddots & 1 \end{vmatrix}$$

$$-u_0 + 2u_1 - u_2 + S\sqrt{u_1}h^2 = 0$$

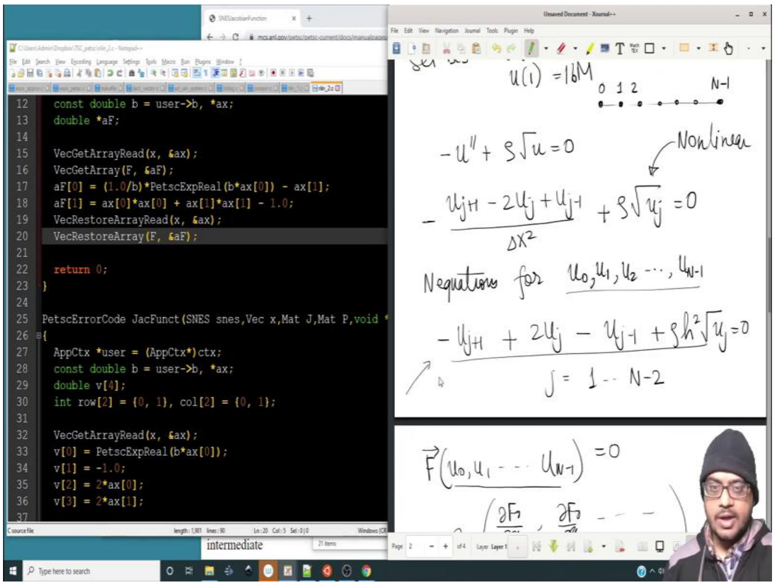
$$\frac{\partial F_1}{\partial u_0} = -1 \quad \frac{\partial F_1}{\partial u_1} = 2 + \frac{Sh^2}{2\sqrt{u_1}}$$

$$\partial F_1/\partial u_2 = -1 \quad \cdots$$

SNES

Hi guys, in this particular lecture we are going to continue on our journey of solving for the steady state distribution of this particular equation right.

(Refer Slide Time: 00:43)



```
  const double b = user->b, *ax;
  double *aF;

  VecGetArrayRead(x, &ax);
  VecGetArray(F, &aF);
  aF[0] = (1.0/b)*PetscExpReal(b*ax[0]) - ax[1];
  aF[1] = ax[0]*ax[0] + ax[1]*ax[1] - 1.0;
  VecRestoreArrayRead(x, &ax);
  VecRestoreArray(F, &aF);

  return 0;
}

PetscErrorCode JacFunct(SNES snes,Vec x,Mat J,Mat P,void *
{
  AppCtx *user = (AppCtx*)ctx;
  const double b = user->b, *ax;
  double v[4];
  int row[2] = {0, 1}, col[2] = {0, 1};

  VecGetArrayRead(x, &ax);
  v[0] = PetscExpReal(b*ax[0]);
  v[1] = -1.0;
  v[2] = 2*ax[0];
  v[3] = 2*ax[1];
```

$u(1) = 16M$

$$-u'' + S\sqrt{u} = 0$$

Nonlinear

$$-\frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} + S\sqrt{u_j} = 0$$

N equations for $u_0, u_1, u_2 \cdots, u_{N-1}$

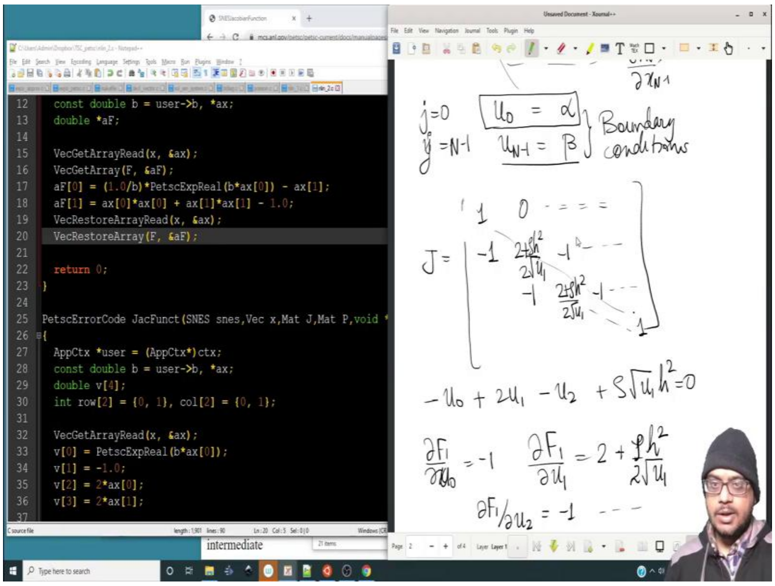
$$-u_{j+1} + 2u_j - u_{j-1} + Sh^2\sqrt{u_j} = 0$$

$$j = 1 \cdots N-2$$

$$\vec{F}(u_0, u_1 \cdots u_{N-1}) = 0$$

So, just to recap this is the kind of discretization we had and the Jacobian would look something like this alright.

(Refer Slide Time: 00:56)



```
  const double b = user->b, *ax;
  double *aF;

  VecGetArrayRead(x, &ax);
  VecGetArray(F, &aF);
  aF[0] = (1.0/b)*PetscExpReal(b*ax[0]) - ax[1];
  aF[1] = ax[0]*ax[0] + ax[1]*ax[1] - 1.0;
  VecRestoreArrayRead(x, &ax);
  VecRestoreArray(F, &aF);

  return 0;
}

PetscErrorCode JacFunct(SNES snes,Vec x,Mat J,Mat P,void *
{
  AppCtx *user = (AppCtx*)ctx;
  const double b = user->b, *ax;
  double v[4];
  int row[2] = {0, 1}, col[2] = {0, 1};

  VecGetArrayRead(x, &ax);
  v[0] = PetscExpReal(b*ax[0]);
  v[1] = -1.0;
  v[2] = 2*ax[0];
  v[3] = 2*ax[1];
```

∂x_{N-1}

$j=0 \quad \boxed{u_0 = \alpha}$

$j=N-1 \quad \underline{u_{N-1} = \beta}$ } Boundary conditions

$$J = \begin{bmatrix} 1 & 0 & \cdots & & \\ -1 & 2+\frac{\rho h^2}{2\sqrt{u_1}} & -1 & \cdots & \\ & -1 & 2+\frac{\rho h^2}{2\sqrt{u_1}} & -1 & \cdots \\ & & & \ddots & 1 \end{bmatrix}$$

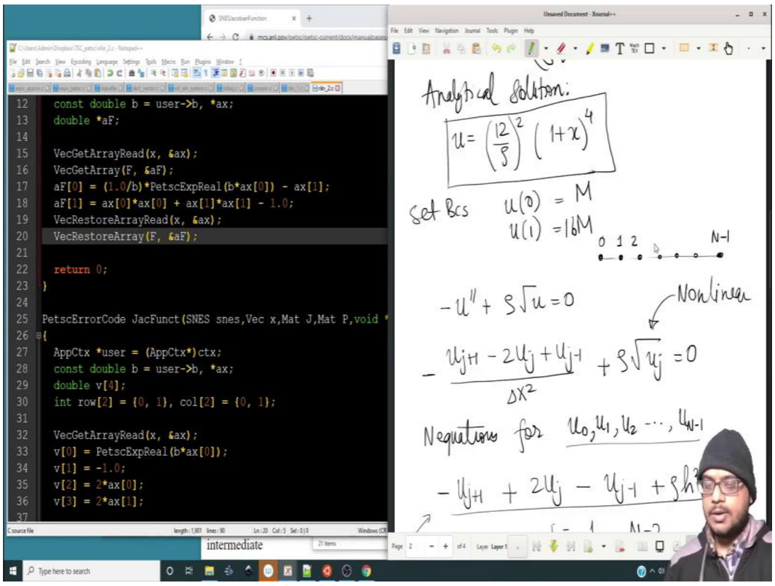
$$-u_0 + 2u_1 - u_2 + \rho\sqrt{u_1}\,h^2 = 0$$

$$\frac{\partial F_1}{\partial u_0} = -1 \qquad \frac{\partial F_1}{\partial u_1} = 2 + \frac{\rho h^2}{2\sqrt{u_1}}$$

$$\partial F_1/\partial u_2 = -1 \quad \cdots$$

So, let us start coding and because it is a one dimensional problem we do not need any dm - going around over here but, we will need a one d grid something like this right.

(Refer Slide Time: 01:17)



```
  const double b = user->b, *ax;
  double *aF;

  VecGetArrayRead(x, &ax);
  VecGetArray(F, &aF);
  aF[0] = (1.0/b)*PetscExpReal(b*ax[0]) - ax[1];
  aF[1] = ax[0]*ax[0] + ax[1]*ax[1] - 1.0;
  VecRestoreArrayRead(x, &ax);
  VecRestoreArray(F, &aF);

  return 0;
}

PetscErrorCode JacFunct(SNES snes,Vec x,Mat J,Mat P,void *
{
  AppCtx *user = (AppCtx*)ctx;
  const double b = user->b, *ax;
  double v[4];
  int row[2] = {0, 1}, col[2] = {0, 1};

  VecGetArrayRead(x, &ax);
  v[0] = PetscExpReal(b*ax[0]);
  v[1] = -1.0;
  v[2] = 2*ax[0];
  v[3] = 2*ax[1];
```

Analytical solution:

$$\boxed{u = \left(\frac{12}{\rho}\right)^2 (1+x)^4}$$

Set Bcs $u(0) = M$

$u(1) = 16M$

$$-u'' + \rho\sqrt{u} = 0$$

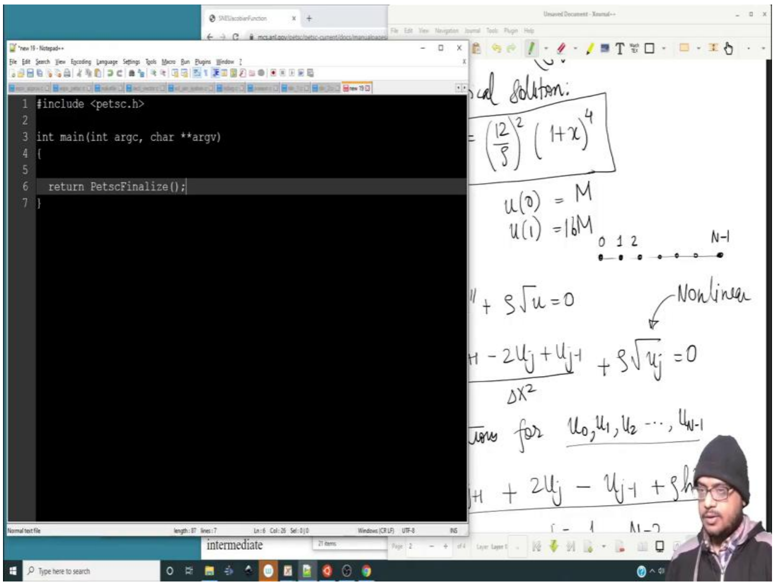
Nonlinear

$$-\frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} + \rho\sqrt{u_j} = 0$$

N equations for $\underline{u_0, u_1, u_2 \cdots, u_{N-1}}$

$$\underline{-u_{j+1} + 2u_j - u_{j-1} + \rho h^2}$$

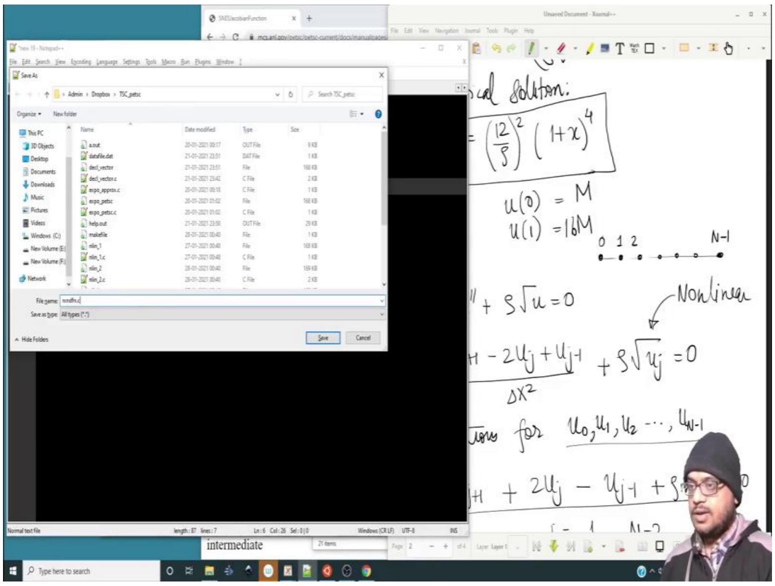
(Refer Slide Time: 01:23)



```
#include <petsc.h>

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

  return PetscFinalize();
}
```

So, let us make a new file. So, first things first #include <petsc.h> int main(int argc, char **argv) return PetscFinalize() alright.

(Refer Slide Time: 01:55)

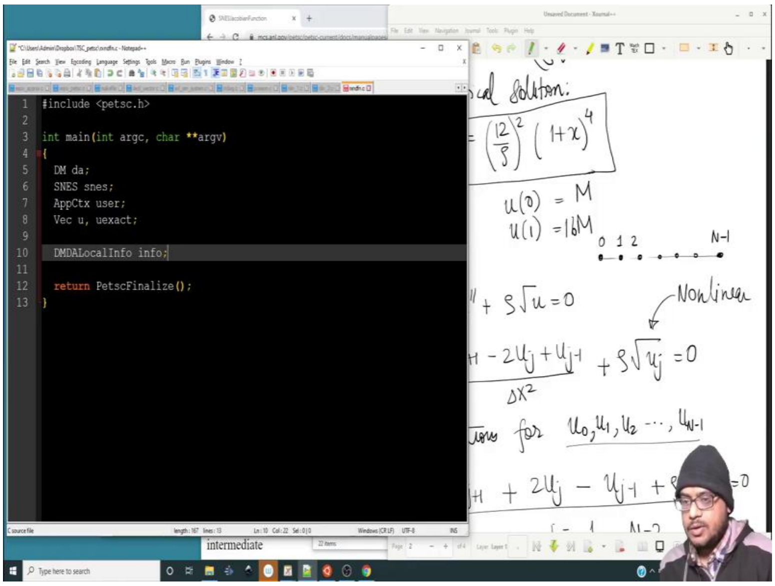


File name: rxndfn.c

Save as type: All types (*.*)

So, let us save this as rxn_dfn.c just to signify its a reaction diffusion system alright. So, first things first we need to define the various things we will need. So, we are going to need a DM da.

(Refer Slide Time: 02:29)



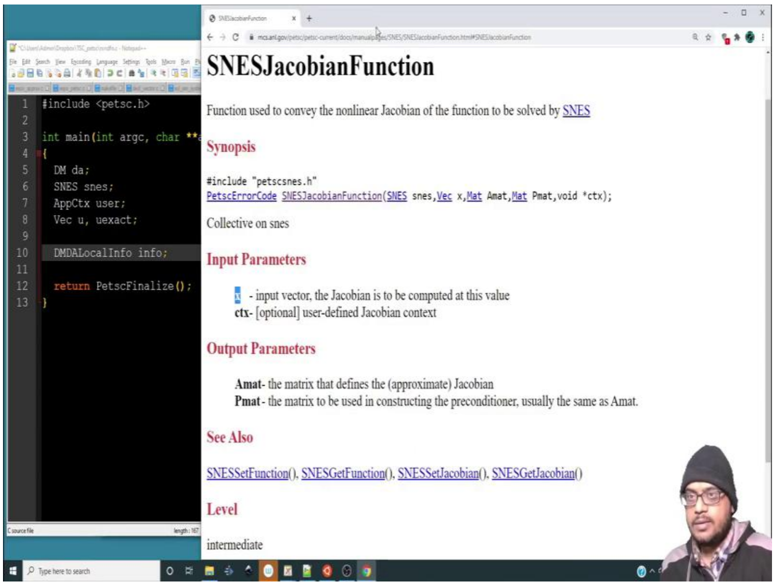
```
#include <petsc.h>

int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;

  DMDALocalInfo info;

  return PetscFinalize();
}
```

cal Solution:
$= \left(\frac{12}{\rho}\right)^2 (1+x)^4$
$u(0) = M$
$u(1) = 16M$
0 1 2 N-1
$'' + \rho\sqrt{u} = 0$
Nonlinear
$\frac{\quad - 2u_j + u_{j-1}}{\Delta x^2} + \rho\sqrt{u_j} = 0$
for $u_0, u_1, u_2 \cdots, u_{N-1}$
$+ 2u_j - u_{j-1} +$ $=0$

But in this case it will be DM and will call it da; and we are going to need a SNES about from this we are going to need AppCtx which is to hold the value of rho and all these things. So, AppCtx we will define it as user because the user is going to supplied, we are going to need two vectors one is u and one is uexact just to make a comparison we are going to need anyway.

So, let us continue on this whenever we need a new variable we will sort of declare it and we need a DMDALocalInfo for performing the loops and we will call it info.

(Refer Slide Time: 03:36)



SNESJacobianFunction

Function used to convey the nonlinear Jacobian of the function to be solved by SNES

Synopsis

```
#include "petscsnes.h"
PetscErrorCode SNESJacobianFunction(SNES snes,Vec x,Mat Amat,Mat Pmat,void *ctx);
```

Collective on snes

Input Parameters

x - input vector, the Jacobian is to be computed at this value
ctx- [optional] user-defined Jacobian context

Output Parameters

Amat- the matrix that defines the (approximate) Jacobian
Pmat - the matrix to be used in constructing the preconditioner, usually the same as Amat.

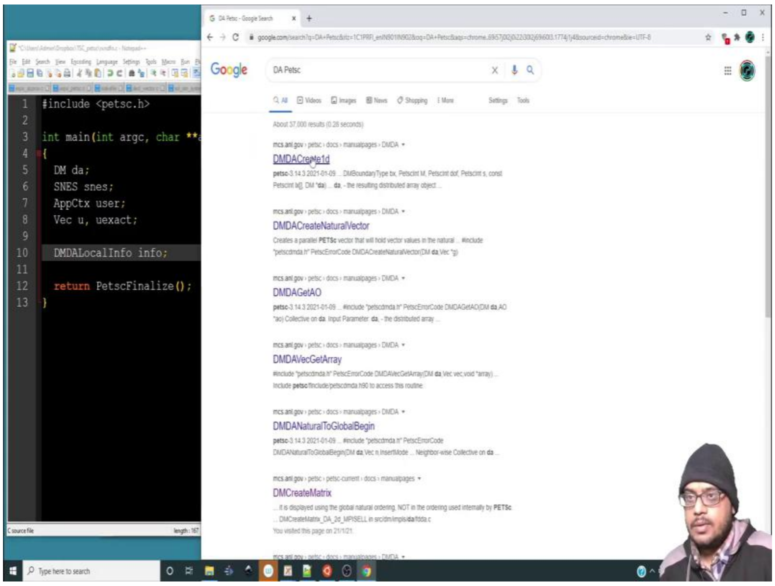
See Also

SNESSetFunction(), SNESGetFunction(), SNESSetJacobian(), SNESGetJacobian()

Level

intermediate

(Refer Slide Time: 03:28)



```
#include <petsc.h>

int main(int argc, char **
{
   DM da;
   SNES snes;
   AppCtx user;
   Vec u, uexact;

   DMDALocalInfo info;

   return PetscFinalize();
}
```

DA Petsc

About 37,000 results (0.28 seconds)

mcs.anl.gov › petsc › docs › manualpages › DMDA
DMDACreate1d
petsc-3.14.3 2021-01-09 ... DMBoundaryType bx, PetscInt M, PetscInt dof, PetscInt s, const PetscInt lx[], DM *da) ... da - the resulting distributed array object ...

mcs.anl.gov › petsc › docs › manualpages › DMDA
DMDACreateNaturalVector
Creates a parallel PETSc vector that will hold vector values in the natural ... #include "petscdmda.h" PetscErrorCode DMDACreateNaturalVector(DM da,Vec *g)

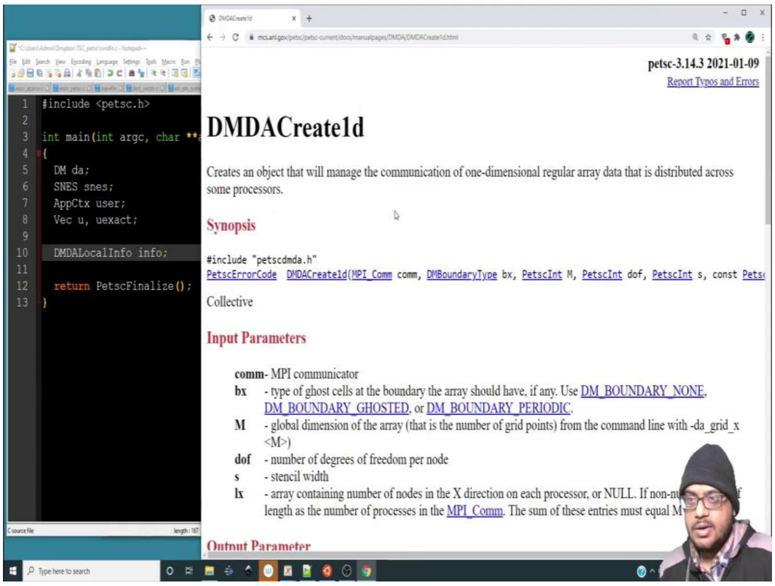
mcs.anl.gov › petsc › docs › manualpages › DMDA
DMDAGetAO
petsc-3.14.3 2021-01-09 ... #include "petscdmda.h" PetscErrorCode DMDAGetAO(DM da,AO *ao) Collective on da. Input Parameter. da - the distributed array ...

mcs.anl.gov › petsc › docs › manualpages › DMDA
DMDAVecGetArray
#include "petscdmda.h" PetscErrorCode DMDAVecGetArray(DM da,Vec vec,void *array) ... Include petsc/finclude/petscdmda.h90 to access this routine.

mcs.anl.gov › petsc › docs › manualpages › DMDA
DMDANaturalToGlobalBegin
petsc-3.14.3 2021-01-09 ... #include "petscdmda.h" PetscErrorCode DMDANaturalToGlobalBegin(DM da,Vec n,InsertMode ... Neighbor-wise Collective on da ...

mcs.anl.gov › petsc › petsc-current › docs › manualpages
DMCreateMatrix
... It is displayed using the global natural ordering, NOT in the ordering used internally by PETSc. ... DMCreateMatrix_DA_2d_MPISELL in src/dm/impls/da/fdda.c
You visited this page on 21/1/21.

(Refer Slide Time: 03:34)



petsc-3.14.3 2021-01-09
Report Typos and Errors

DMDACreate1d

Creates an object that will manage the communication of one-dimensional regular array data that is distributed across some processors.

Synopsis

```
#include "petscdmda.h"
PetscErrorCode  DMDACreate1d(MPI_Comm comm, DMBoundaryType bx, PetscInt M, PetscInt dof, PetscInt s, const Pets
```

Collective

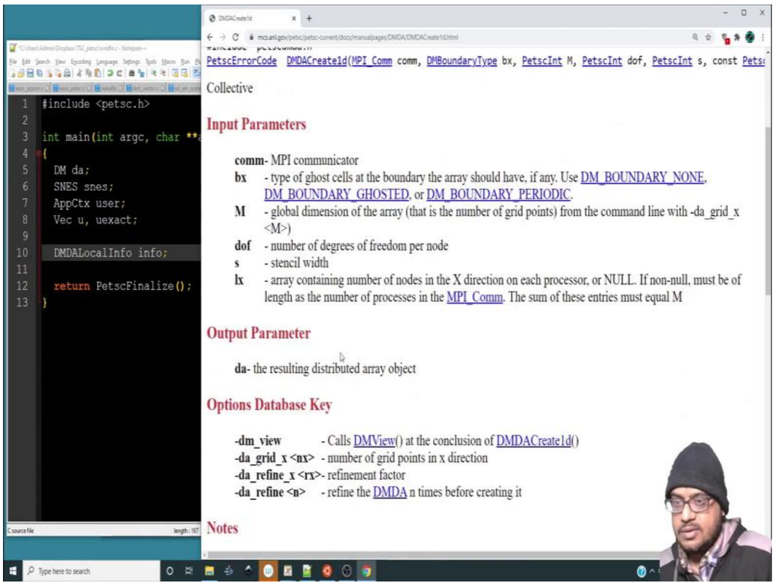
Input Parameters

- **comm**- MPI communicator
- **bx** - type of ghost cells at the boundary the array should have, if any. Use DM_BOUNDARY_NONE, DM_BOUNDARY_GHOSTED, or DM_BOUNDARY_PERIODIC.
- **M** - global dimension of the array (that is the number of grid points) from the command line with -da_grid_x <M>)
- **dof** - number of degrees of freedom per node
- **s** - stencil width
- **lx** - array containing number of nodes in the X direction on each processor, or NULL. If non-nu length as the number of processes in the MPI_Comm. The sum of these entries must equal M

Output Parameter

So, the data type DA is essentially DMDACreate1d ok.

(Refer Slide Time: 03:42)



```
#include <petsc.h>

int main(int argc, char **

{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;

  DMDALocalInfo info;

  return PetscFinalize();
}
```

PetscErrorCode DMDACreate1d(MPI_Comm comm, DMBoundaryType bx, PetscInt M, PetscInt dof, PetscInt s, const Petsc

Collective

Input Parameters

comm- MPI communicator
bx - type of ghost cells at the boundary the array should have, if any. Use DM_BOUNDARY_NONE, DM_BOUNDARY_GHOSTED, or DM_BOUNDARY_PERIODIC.
M - global dimension of the array (that is the number of grid points) from the command line with -da_grid_x <M>)
dof - number of degrees of freedom per node
s - stencil width
lx - array containing number of nodes in the X direction on each processor, or NULL. If non-null, must be of length as the number of processes in the MPI_Comm. The sum of these entries must equal M

Output Parameter

da- the resulting distributed array object

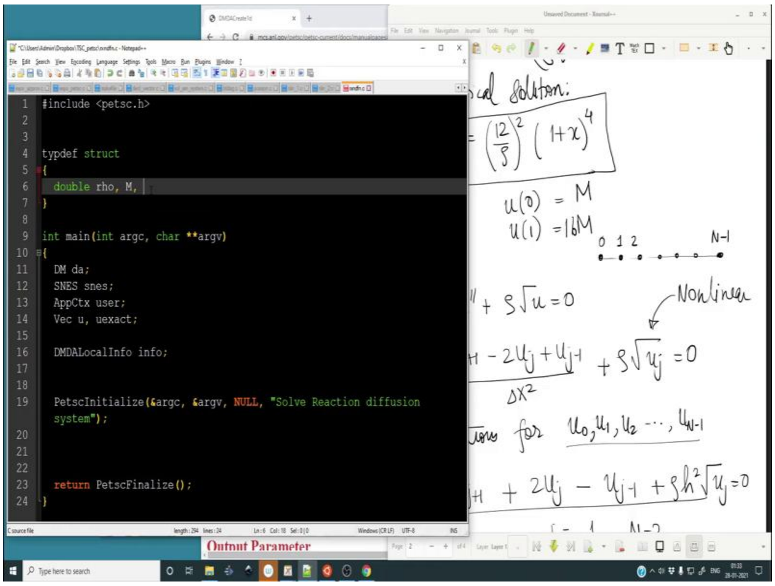
Options Database Key

-dm_view - Calls DMView() at the conclusion of DMDACreate1d()
-da_grid_x <nx> - number of grid points in x direction
-da_refine_x <rx>- refinement factor
-da_refine <n> - refine the DMDA n times before creating it

Notes

So, one dimensional regular array and this is the distributed array object ok. So, we going to use this DMDACreate1d and it will help us in creating the one dimensional grid alright.

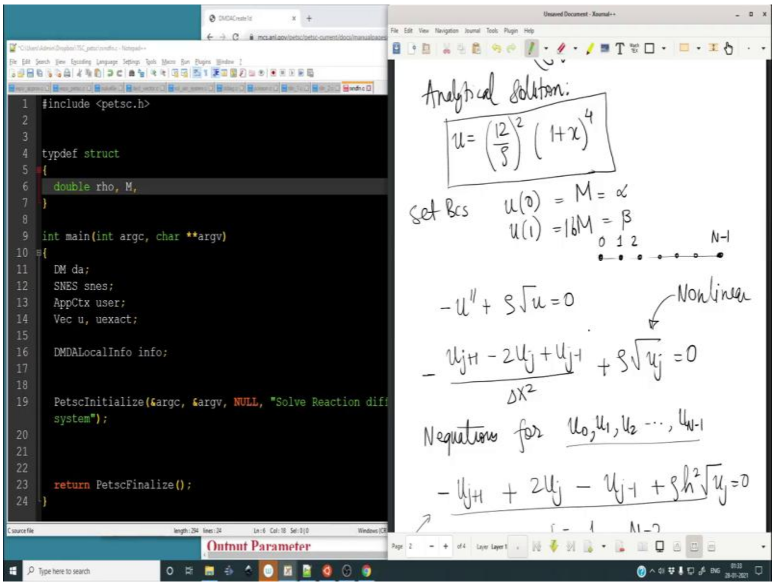
(Refer Slide Time: 03:57)



```
#include <petsc.h>


typdef struct
{
  double rho, M,
}

int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;

  DMDALocalInfo info;


  PetscInitialize(&argc, &argv, NULL, "Solve Reaction diffusion
  system");



  return PetscFinalize();
}
```

So, let us proceed so, we will need PetscInitialize we have PETSC sorry this will have one &argc, &argv, NULL and the help text alright then, we will go ahead and let us first declare what the user context will be so, we have to declare the struct so typdef struct. So, this will be double so what all things do we need so, we going to need rho we going to need M; so we going to need ρ and M we also going to need α and β just to specify the boundary conditions in terms of M alright.

(Refer Slide Time: 05:13)



```
#include <petsc.h>


typdef struct
{
  double rho, M,
}

int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;

  DMDALocalInfo info;


  PetscInitialize(&argc, &argv, NULL, "Solve Reaction diff
  system");



  return PetscFinalize();
}
```

Analytical Solution:

$$\boxed{u = \left(\frac{12}{\rho}\right)^2 (1+x)^4}$$

Set Bcs $u(0) = M = \alpha$

$u(1) = 16M = \beta$

$-u'' + \rho\sqrt{u} = 0$ Nonlinear

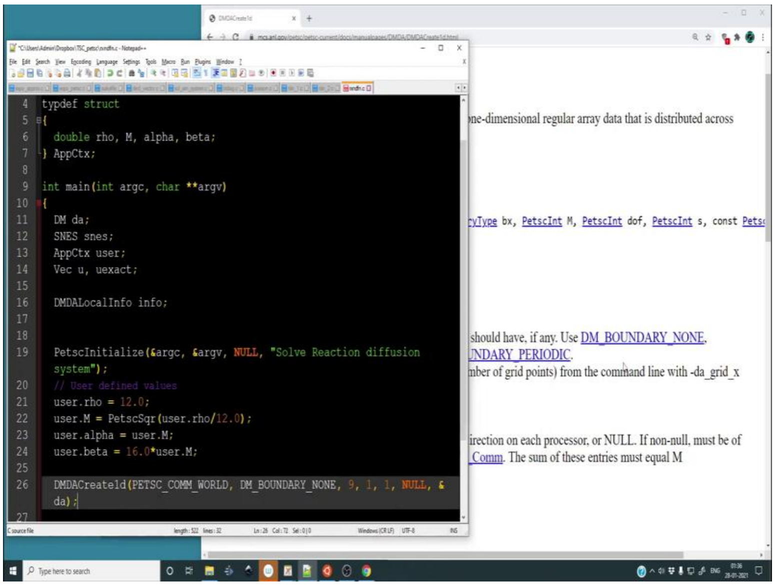
$$-\frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} + \rho\sqrt{u_j} = 0$$

N equations for $u_0, u_1, u_2 \cdots, u_{N-1}$

$-u_{j+1} + 2u_j - u_{j-1} + \rho h^2\sqrt{u_j} = 0$

Output Parameter

(Refer Slide Time: 05:20)



```
typdef struct
{
  double rho, M, alpha, beta;
} AppCtx;

int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;

  DMDALocalInfo info;


  PetscInitialize(&argc, &argv, NULL, "Solve Reaction diffusion
  system");
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.rho/12.0);
  user.alpha = user.M;
  user.beta = 16.0*user.M;

  DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &
  da);
```

one-dimensional regular array data that is distributed across

ryType bx, PetscInt M, PetscInt dof, PetscInt s, const Petsc

should have, if any. Use DM_BOUNDARY_NONE,
UNDARY_PERIODIC.
mber of grid points) from the command line with -da_grid_x

irection on each processor, or NULL. If non-null, must be of
Comm. The sum of these entries must equal M

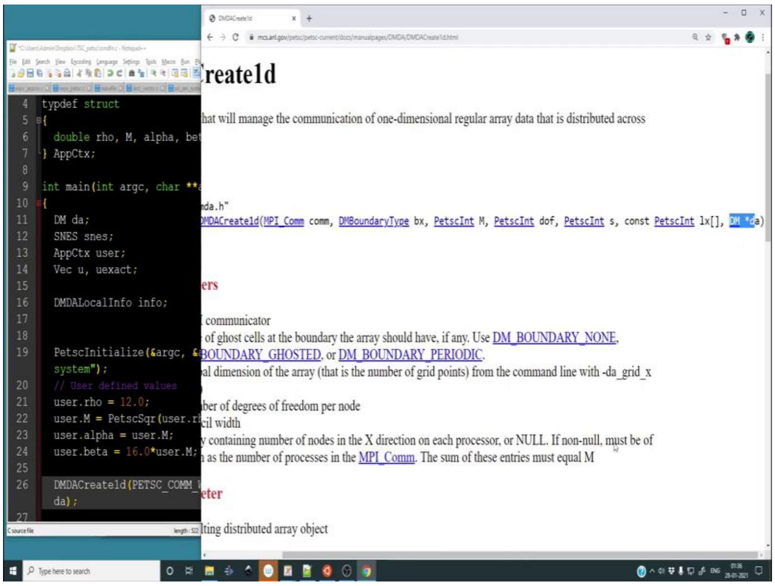
So, let us declare that and that is pretty much it. So, we will define it as AppCtx so that is the data type alright so; let us declare the data type over here so we have user.rho is equal to so what was ρ, ρ in terms of rather M was in terms of ρ right. So, let us define ρ to be 12 that way M becomes 1 so, we can do this to the 12.0 alright; user.M will be the square of user.rho/12.0, 2 well its a square so, we do not need to do this here.

Then, we need to define the α so, user.alpha will be simply user.M, user.beta will be simply 16.0*user.M so, we have defined all the parameters there are alright. So, these are what is required after initialization so, this is these are just the user defined values.

So, once we have; once we have the user defined values we will have DMDA creation so DMDACreate 1d, it will require the communicator PETSC COMM WORLD then DM_BOUNDARY_NONE. So, let me just show you the things that we require the communicator the boundary type, the dimension of the array.

So, let us declare 9 grid points its fine then what do we have number of degrees of freedom stencil width. So, number of degrees of freedom is 1 stencil width is 1 then, number of nodes on each processor we will declare it to the NULL because the PETSC will do the load balancing in case we have multiple processors and lastly we must pass the da, that is the distributed array over here ok.

(Refer Slide Time: 08:34)



```
typdef struct
{
  double rho, M, alpha, bet
} AppCtx;

int main(int argc, char **a
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;

  DMDALocalInfo info;


  PetscInitialize(&argc, &
  system");
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.r
  user.alpha = user.M;
  user.beta = 16.0*user.M;

  DMDACreate1d(PETSC_COMM_W
  da);
```

reate1d

hat will manage the communication of one-dimensional regular array data that is distributed across

```
da.h"
DMDACreate1d(MPI_Comm comm, DMBoundaryType bx, PetscInt M, PetscInt dof, PetscInt s, const PetscInt lx[], DM *da)
```

ers

l communicator
of ghost cells at the boundary the array should have, if any. Use DM_BOUNDARY_NONE, BOUNDARY_GHOSTED, or DM_BOUNDARY_PERIODIC.
al dimension of the array (that is the number of grid points) from the command line with -da_grid_x

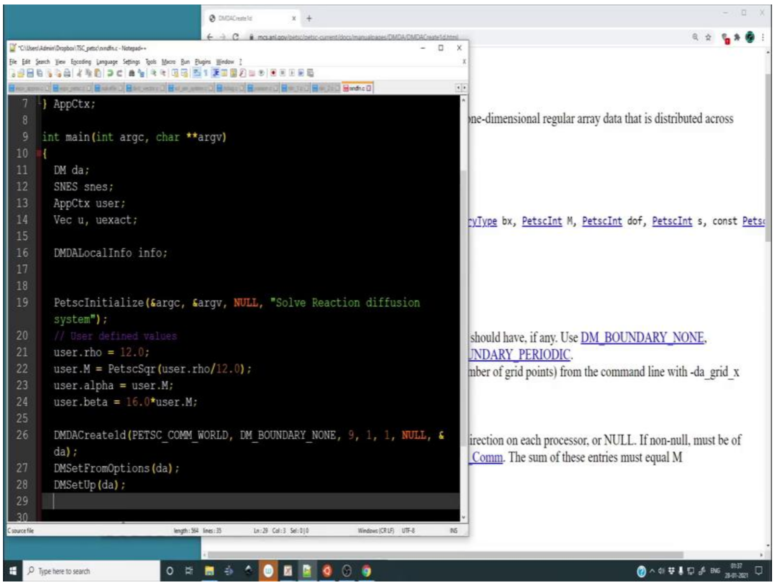
ber of degrees of freedom per node
cil width
y containing number of nodes in the X direction on each processor, or NULL. If non-null, must be of
h as the number of processes in the MPI_Comm. The sum of these entries must equal M

eter

lting distributed array object

So stencil is one because we are defining the h on our own right so, this is it this is how the one dimensional grid is created then, we must do the other thing so DMSetFromOptions we have to pass da then, DMSetUp da and then we have to set up the application context.

(Refer Slide Time: 08:55)



```
} AppCtx;

int main(int argc, char **argv)
{
   DM da;
   SNES snes;
   AppCtx user;
   Vec u, uexact;

   DMDALocalInfo info;


   PetscInitialize(&argc, &argv, NULL, "Solve Reaction diffusion
   system");
   // User defined values
   user.rho = 12.0;
   user.M = PetscSqr(user.rho/12.0);
   user.alpha = user.M;
   user.beta = 16.0*user.M;

   DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &
   da);
   DMSetFromOptions(da);
   DMSetUp(da);
```

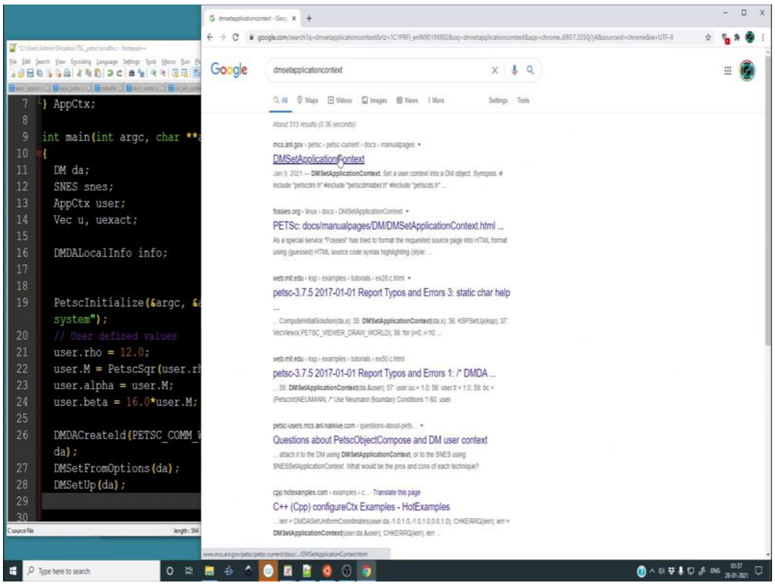
one-dimensional regular array data that is distributed across

ryType bx, PetscInt M, PetscInt dof, PetscInt s, const Petsc

should have, if any. Use DM_BOUNDARY_NONE,
UNDARY_PERIODIC.
mber of grid points) from the command line with -da_grid_x

irection on each processor, or NULL. If non-null, must be of
_Comm. The sum of these entries must equal M

(Refer Slide Time: 09:23)



```
} AppCtx;

int main(int argc, char **a

   DM da;
   SNES snes;
   AppCtx user;
   Vec u, uexact;

   DMDALocalInfo info;


   PetscInitialize(&argc, &a
   system");
   // User defined values
   user.rho = 12.0;
   user.M = PetscSqr(user.rh
   user.alpha = user.M;
   user.beta = 16.0*user.M;

   DMDACreate1d(PETSC_COMM_W
   da);
   DMSetFromOptions(da);
   DMSetUp(da);
```

google.com/search?q=dmsetapplicationcontext

dmsetapplicationcontext

About 313 results (0.36 seconds)

mcs.anl.gov › petsc › petsc-current › docs › manualpages
DMSetApplicationContext
Jan 9, 2021 — DMSetApplicationContext. Set a user context into a DM object. Synopsis. #include "petscdm.h" #include "petscdmlabel.h" #include "petscds.h" ...

fossies.org › linux › docs › DMSetApplicationContext
PETSc: docs/manualpages/DM/DMSetApplicationContext.html ...
As a special service "Fossies" has tried to format the requested source page into HTML format using (guessed) HTML source code syntax highlighting (style: ...

web.mit.edu › ksp › examples › tutorials › ex28.c.html
petsc-3.7.5 2017-01-01 Report Typos and Errors 3: static char help ...
... ComputeInitialSolution(da,x); 35: DMSetApplicationContext(da,x); 36: KSPSetUp(ksp); 37: VecView(x,PETSC_VIEWER_DRAW_WORLD); 38: for (i=0; i<10; ...

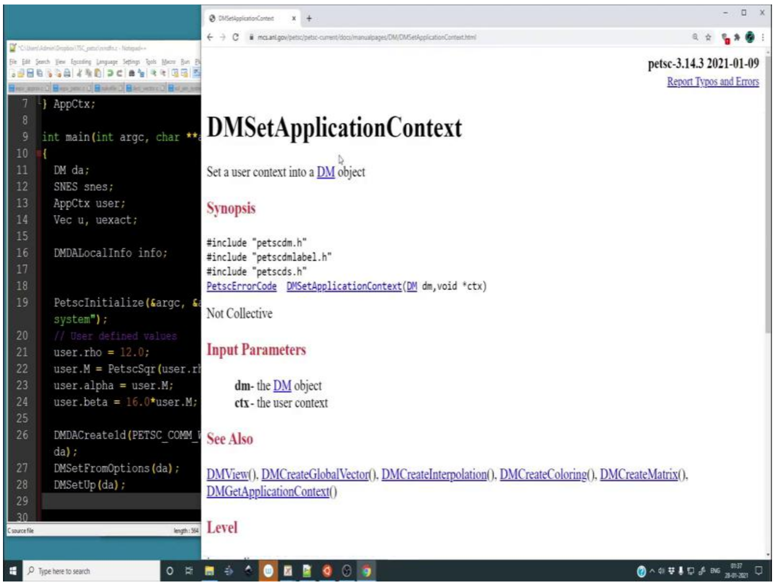
web.mit.edu › ksp › examples › tutorials › ex50.c.html
petsc-3.7.5 2017-01-01 Report Typos and Errors 1: /* DMDA ...
... 55: DMSetApplicationContext(da,&user); 57: user.uu = 1.0; 58: user.tt = 1.0; 59: bc = (PetscInt)NEUMANN; /* Use Neumann Boundary Conditions */ 60: user.

petsc-users.mcs.anl.narkive.com › questions-about-pets...
Questions about PetscObjectCompose and DM user context
... attach it to the DM using DMSetApplicationContext, or to the SNES using SNESSetApplicationContext. What would be the pros and cons of each technique?

cpp.hotexamples.com › examples › c... · Translate this page
C++ (Cpp) configureCtx Examples - HotExamples
... ierr = DMDASetUniformCoordinates(user.da,-1.0,1.0,-1.0,1.0,0.0,1.0); CHKERRQ(ierr); ierr = DMSetApplicationContext(user.da,&user); CHKERRQ(ierr); ierr ...

So, the there is a function DMSetApplicationContext. So, we must sort of tell the distributed array that whatever the variable or all the parameters are it has to be distributed over the grid. I mean it is not required for this problem, but there may be a problem with it with this would be required.

(Refer Slide Time: 09:30)



```
} AppCtx;

int main(int argc, char **a
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;

  DMDALocalInfo info;


  PetscInitialize(&argc, &a
  system");
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.rh
  user.alpha = user.M;
  user.beta = 16.0*user.M;

  DMDACreate1d(PETSC_COMM_W
  da);
  DMSetFromOptions(da);
  DMSetUp(da);
```

petsc-3.14.3 2021-01-09

Report Typos and Errors

DMSetApplicationContext

Set a user context into a DM object

Synopsis

```
#include "petscdm.h"
#include "petscdmlabel.h"
#include "petscds.h"
PetscErrorCode  DMSetApplicationContext(DM dm,void *ctx)
```

Not Collective

Input Parameters

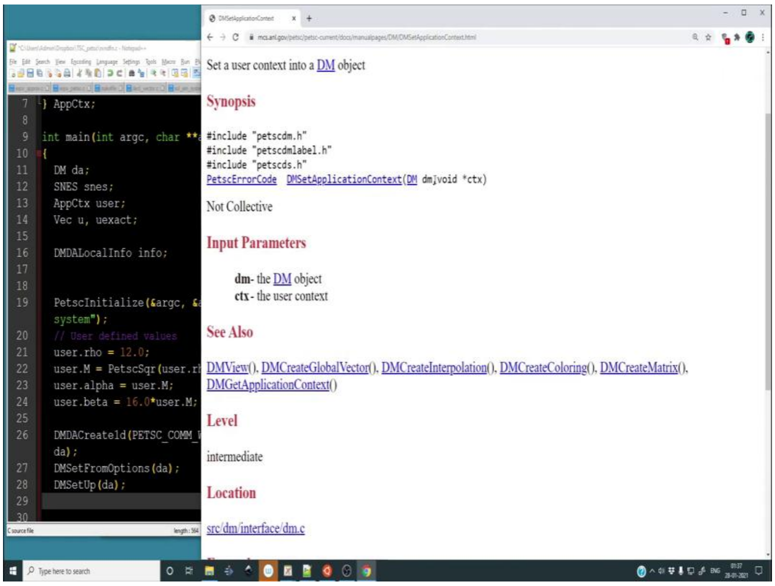
dm- the DM object
ctx - the user context

See Also

DMView(), DMCreateGlobalVector(), DMCreateInterpolation(), DMCreateColoring(), DMCreateMatrix(), DMGetApplicationContext()

Level

(Refer Slide Time: 09:31)



```
} AppCtx;

int main(int argc, char **a
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;

  DMDALocalInfo info;


  PetscInitialize(&argc, &a
  system");
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.rh
  user.alpha = user.M;
  user.beta = 16.0*user.M;

  DMDACreate1d(PETSC_COMM_W
  da);
  DMSetFromOptions(da);
  DMSetUp(da);
```

Set a user context into a DM object

Synopsis

```
#include "petscdm.h"
#include "petscdmlabel.h"
#include "petscds.h"
PetscErrorCode  DMSetApplicationContext(DM dm,void *ctx)
```

Not Collective

Input Parameters

dm- the DM object
ctx - the user context

See Also

DMView(), DMCreateGlobalVector(), DMCreateInterpolation(), DMCreateColoring(), DMCreateMatrix(), DMGetApplicationContext()

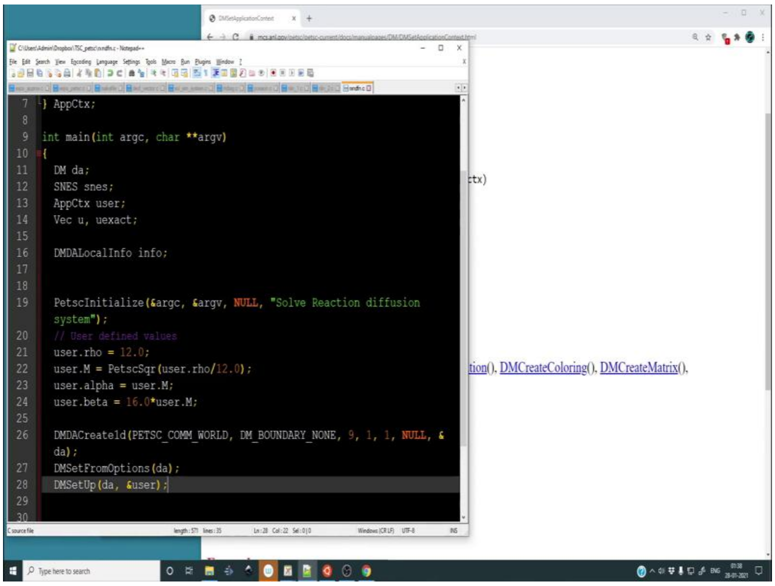
Level

intermediate

Location

src/dm/interface/dm.c

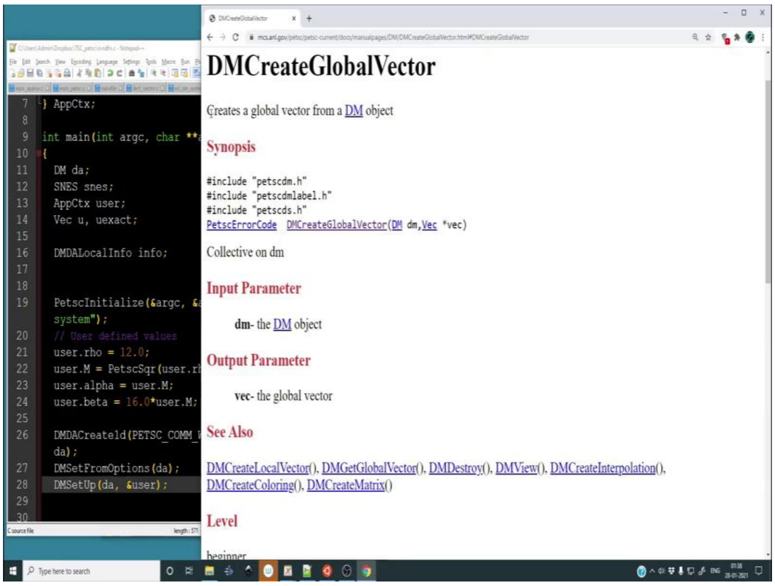
(Refer Slide Time: 09:51)



```
} AppCtx;

int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;

  DMDALocalInfo info;


  PetscInitialize(&argc, &argv, NULL, "Solve Reaction diffusion system");
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.rho/12.0);
  user.alpha = user.M;
  user.beta = 16.0*user.M;

  DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &da);
  DMSetFromOptions(da);
  DMSetUp(da, &user);
```

So, we must pass to the array the address of the user defined context variables ok because, if it is not done it will not be able to see that struct that we have defined like this right ok. So, now that we have this we will create a global vector.

(Refer Slide Time: 10:18)



```
} AppCtx;

int main(int argc, char **a

  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;

  DMDALocalInfo info;


  PetscInitialize(&argc, &a
  system");
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.rh
  user.alpha = user.M;
  user.beta = 16.0*user.M;

  DMDACreate1d(PETSC_COMM_W
  da);
  DMSetFromOptions(da);
  DMSetUp(da, &user);
```

DMCreateGlobalVector

Creates a global vector from a DM object

Synopsis

```
#include "petscdm.h"
#include "petscdmlabel.h"
#include "petscds.h"
PetscErrorCode  DMCreateGlobalVector(DM dm,Vec *vec)
```

Collective on dm

Input Parameter

dm- the DM object

Output Parameter

vec- the global vector

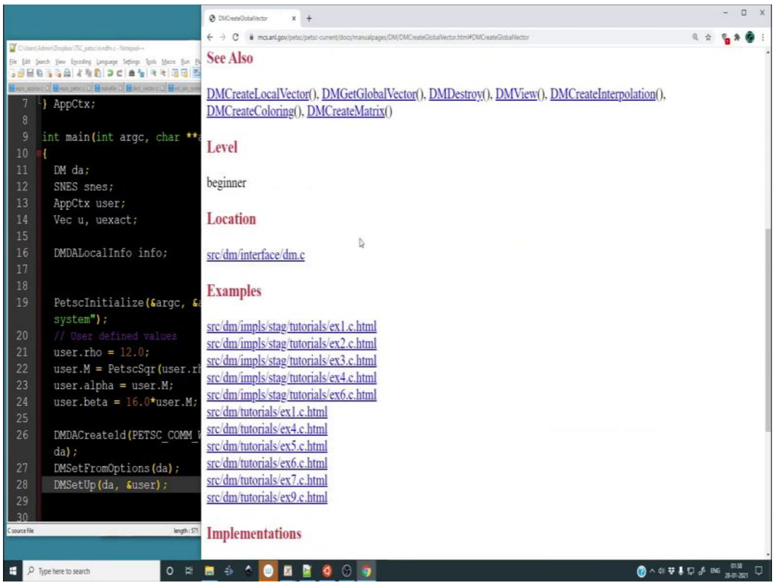
See Also

DMCreateLocalVector(), DMGetGlobalVector(), DMDestroy(), DMView(), DMCreateInterpolation(), DMCreateColoring(), DMCreateMatrix()

Level

beginner

(Refer Slide Time: 10:22)



```
} AppCtx;

int main(int argc, char **

{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;

  DMDALocalInfo info;


  PetscInitialize(&argc, &
  system");
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.r
  user.alpha = user.M;
  user.beta = 16.0*user.M;

  DMDACreate1d(PETSC_COMM_
  da);
  DMSetFromOptions(da);
  DMSetUp(da, &user);
```

See Also

DMCreateLocalVector(), DMGetGlobalVector(), DMDestroy(), DMView(), DMCreateInterpolation(), DMCreateColoring(), DMCreateMatrix()

Level

beginner

Location

src/dm/interface/dm.c

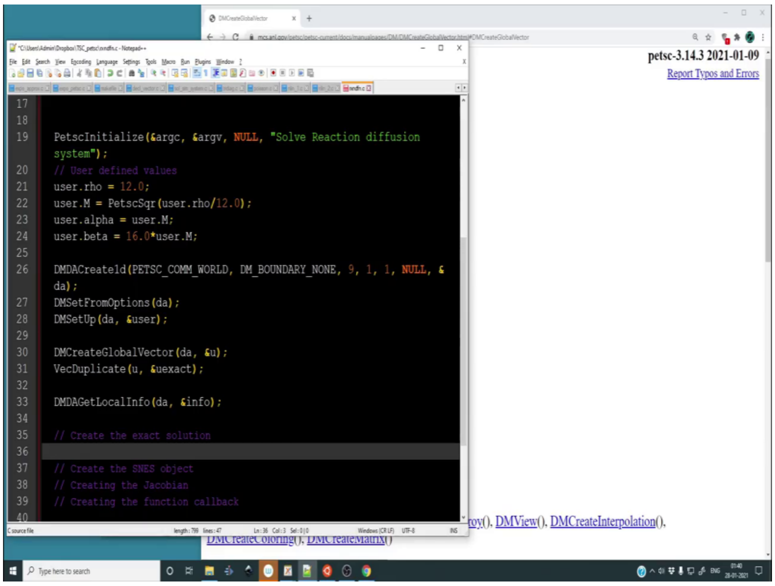
Examples

src/dm/impls/stag/tutorials/ex1.c.html
src/dm/impls/stag/tutorials/ex2.c.html
src/dm/impls/stag/tutorials/ex3.c.html
src/dm/impls/stag/tutorials/ex4.c.html
src/dm/impls/stag/tutorials/ex6.c.html
src/dm/tutorials/ex1.c.html
src/dm/tutorials/ex4.c.html
src/dm/tutorials/ex5.c.html
src/dm/tutorials/ex6.c.html
src/dm/tutorials/ex7.c.html
src/dm/tutorials/ex9.c.html

Implementations

So, let me show this. So, this is the command which will create a global vector on the grid da. So, it is the same as declaring the sort of associativity of u on d.

(Refer Slide Time: 10:36)



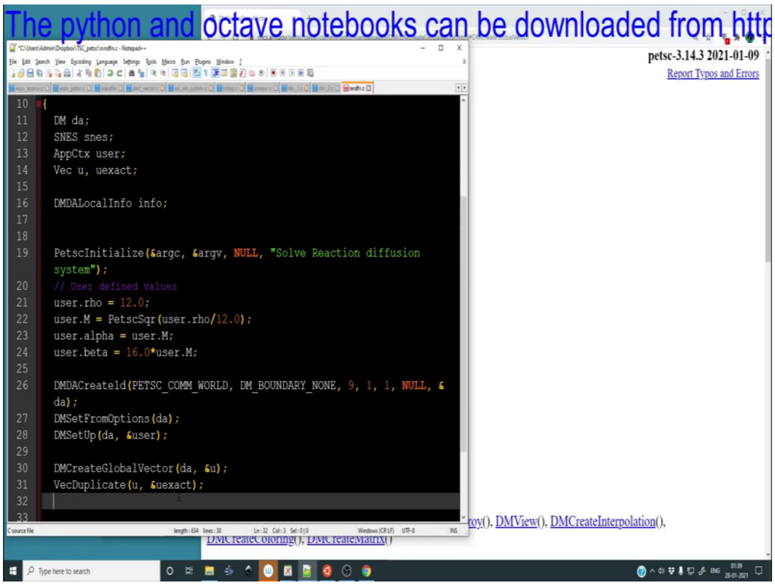
```


  PetscInitialize(&argc, &argv, NULL, "Solve Reaction diffusion
  system");
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.rho/12.0);
  user.alpha = user.M;
  user.beta = 16.0*user.M;

  DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &
  da);
  DMSetFromOptions(da);
  DMSetUp(da, &user);

  DMCreateGlobalVector(da, &u);
  VecDuplicate(u, &uexact);

  DMDAGetLocalInfo(da, &info);

  // Create the exact solution

  // Create the SNES object
  // Creating the Jacobian
  // Creating the function callback
```

petsc-3.14.3 2021-01-09
Report Typos and Errors

roy(), DMView(), DMCreateInterpolation(),
DMCreateColoring(), DMCreateMatrix()

So, DMCreateGlobalVector da and the address of u gets passed now, we will duplicate this vector Vec so we could have simply done create vector but that is ok, Duplicate u into uexact alright.

(Refer Slide Time: 10:55)

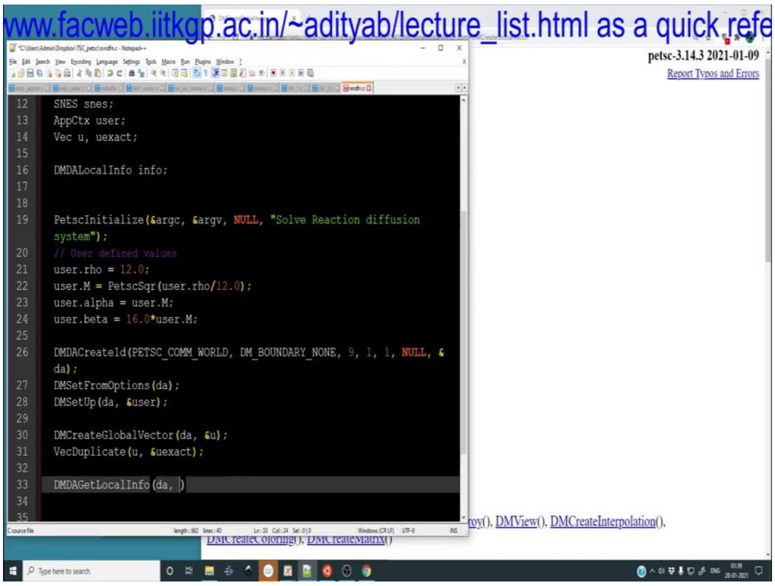


The python and octave notebooks can be downloaded from http

```
{
    DM da;
    SNES snes;
    AppCtx user;
    Vec u, uexact;

    DMDALocalInfo info;


    PetscInitialize(&argc, &argv, NULL, "Solve Reaction diffusion system");
    // User defined values
    user.rho = 12.0;
    user.M = PetscSqr(user.rho/12.0);
    user.alpha = user.M;
    user.beta = 16.0*user.M;

    DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &da);
    DMSetFromOptions(da);
    DMSetUp(da, &user);

    DMCreateGlobalVector(da, &u);
    VecDuplicate(u, &uexact);

```

petsc-3.14.3 2021-01-09
Report Typos and Errors
roy(), DMView(), DMCreateInterpolation(),
DMCreateColoring(), DMCreateMatrix()

(Refer Slide Time: 11:15)



www.facweb.iitkgp.ac.in/~adityab/lecture_list.html as a quick refe

```
    SNES snes;
    AppCtx user;
    Vec u, uexact;

    DMDALocalInfo info;


    PetscInitialize(&argc, &argv, NULL, "Solve Reaction diffusion system");
    // User defined values
    user.rho = 12.0;
    user.M = PetscSqr(user.rho/12.0);
    user.alpha = user.M;
    user.beta = 16.0*user.M;

    DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &da);
    DMSetFromOptions(da);
    DMSetUp(da, &user);

    DMCreateGlobalVector(da, &u);
    VecDuplicate(u, &uexact);

    DMDAGetLocalInfo(da, )

```

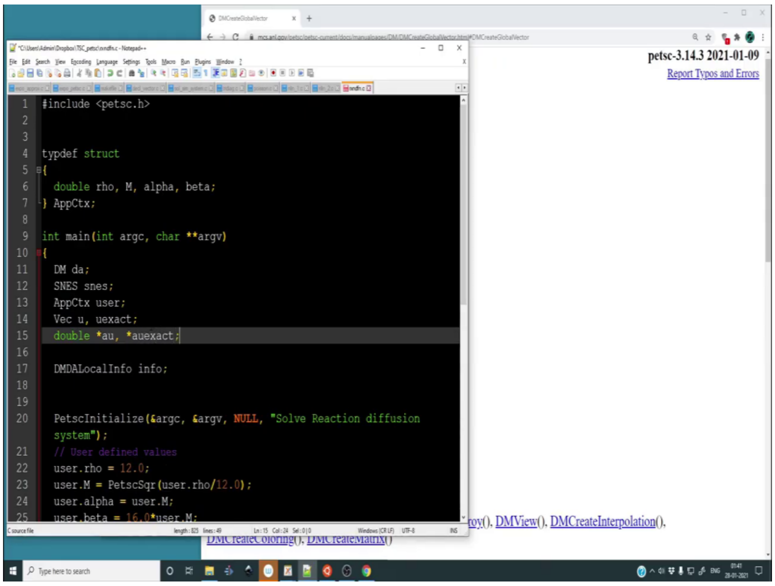
petsc-3.14.3 2021-01-09
Report Typos and Errors
roy(), DMView(), DMCreateInterpolation(),
DMCreateColoring(), DMCreateMatrix()

Then, what we need to do is fetch the rather before fetching what we will do is because we need to pass this into the array so, DMDAGetLocalInfo we will get it into variable info alright. So, this is just because we going to pass info to the arrays because, we need to loop over the grid points. So, we will eventually need this ok.

So then, what should we do? So, our (Refer Time: 12:04) should consist of Create the exact solution because we will have to compare it eventually alright then, we must Create the SNES object. So, creating the SNES object will require Creating the Jacobian as well; and it will require Creating the function callback. Well, in this case we will need all this we will need to create the function callback and we will need to create the Jacobian as well alright.

So let us first create the exact solution so, what do we need. So, let us first allocate let us first create the auxiliary Vec.

(Refer Slide Time: 13:22)

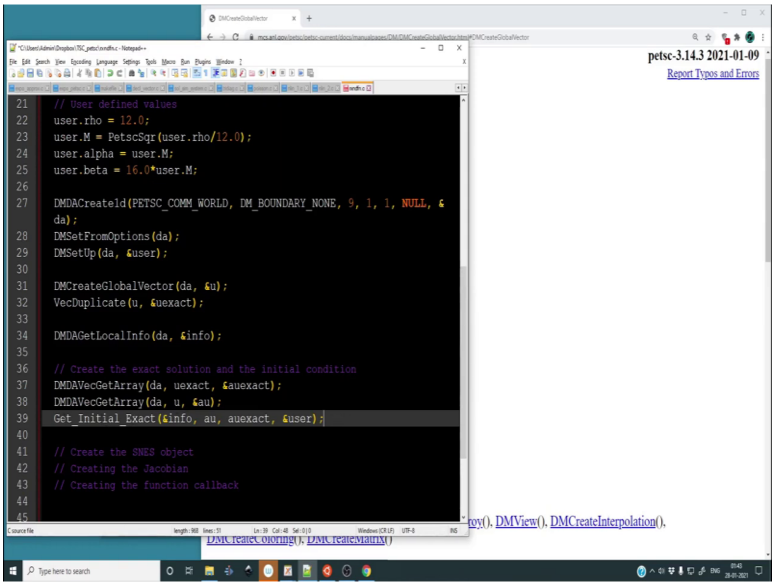


```
#include <petsc.h>


typdef struct
{
  double rho, M, alpha, beta;
} AppCtx;

int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;
  double *au, *auexact;

  DMDALocalInfo info;


  PetscInitialize(&argc, &argv, NULL, "Solve Reaction diffusion
  system");
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.rho/12.0);
  user.alpha = user.M;
  user.beta = 16.0*user.M;
```

petsc-3.14.3 2021-01-09
Report Typos and Errors
roy(), DMView(), DMCreateInterpolation(),
DMCreateColoring(), DMCreateMatrix()

So, to hold the nodal values we will create the auxiliary function so double *au, and *auexact.

(Refer Slide Time: 13:35)



```
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.rho/12.0);
  user.alpha = user.M;
  user.beta = 16.0*user.M;

  DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &
  da);
  DMSetFromOptions(da);
  DMSetUp(da, &user);

  DMCreateGlobalVector(da, &u);
  VecDuplicate(u, &uexact);

  DMDAGetLocalInfo(da, &info);

  // Create the exact solution and the initial condition
  DMDAVecGetArray(da, uexact, &auexact);
  DMDAVecGetArray(da, u, &au);
  Get_Initial_Exact(&info, au, auexact, &user);

  // Create the SNES object
  // Creating the Jacobian
  // Creating the function callback


```

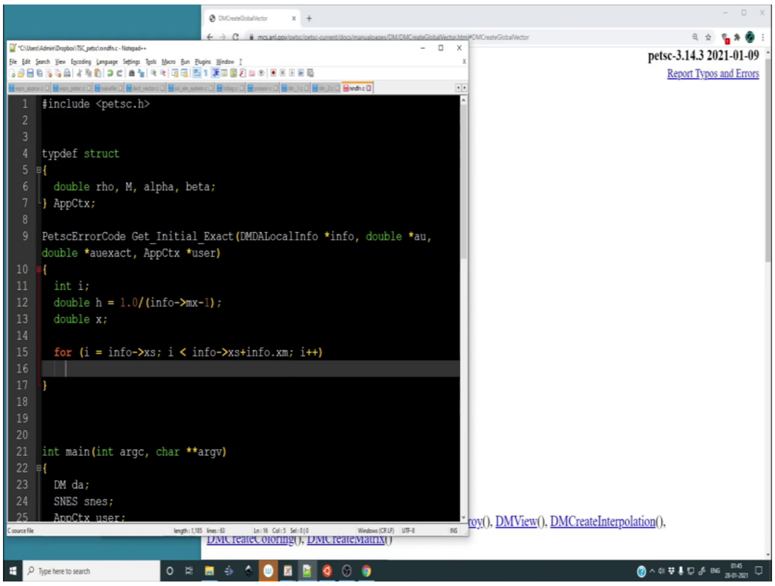
petsc-3.14.3 2021-01-09
Report Typos and Errors
roy(), DMView(), DMCreateInterpolation(),
DMCreateColoring(), DMCreateMatrix()

And then, over here we will do Vec or not Vec DMDAVecGetArray from da uexact needs to be put into auexact alright. Apart from this, what do we need? That is all; that is all that we need? In fact, we need to pause u so that we can create the initial condition so, we will make a single function which will Create the exact solution and the initial condition we can do it in one way.

So, DMDAVecGetArray from da get u and pass it to au. So, now that we have the objects or rather the we have assigned u exact to auexact and u to au we can pass them to GetInitialConditions and Get Initial and Exact; and we will pass to it the address of the info and we will pass the auxiliary matrices and we will pass the user context because, to construct the exact solution we will also need the user context.

So, what we can do is pass the address of user so, we will keep this form and create this function so Get Initial Exact. And we can copy this and copy this and we can create the function over here.

(Refer Slide Time: 15:57)



```
#include <petsc.h>


typdef struct
{
  double rho, M, alpha, beta;
} AppCtx;

PetscErrorCode Get_Initial_Exact(DMDALocalInfo *info, double *au,
double *auexact, AppCtx *user)
{
  int i;
  double h = 1.0/(info->mx-1);
  double x;

  for (i = info->xs; i < info->xs+info.xm; i++)
    
}



int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
```

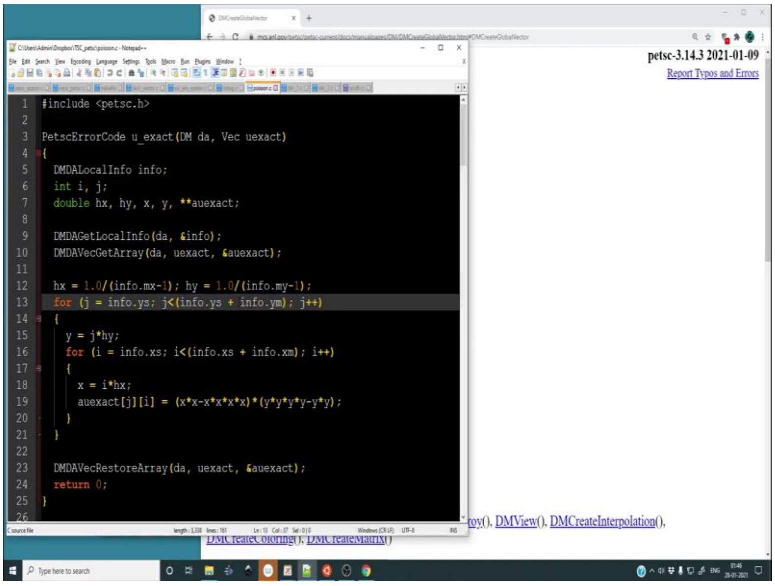
petsc-3.14.3 2021-01-09
Report Typos and Errors
roy(), DMView(), DMCreateInterpolation(),
DMCreateColoring(), DMCreateMatrix()

So, the output will be Petsc Error Code. So, this is the function so, info instead of this we will have DMDALocalInfo star info, we will have double *au and double *auexact and void or in fact AppCtx *user because, we do not want to type cast it inside and we can use it directly. In case you put a void star you need to typecast it to AppCtx, we have done this in the previous example we do not need to typecast.

So, inside this we will create int i and double h that is the grid spacing, which will be 1.0 divided by the number of grid points minus 1 and the number of grid points is held inside the info so, 1.0/(info->mx - 1). So, this is the number of grids minus 1 alright. We also need double x for i going from i = info->xs; i < info->xs+info->xm; i++.

And so, where do we get this from and we have used this in one of the previous solver is basically this bit of code.

(Refer Slide Time: 18:00)

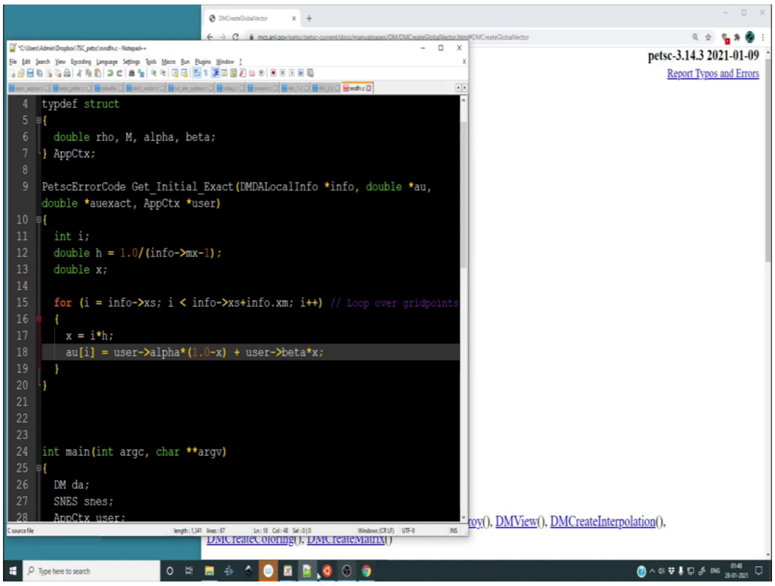


```c
#include <petsc.h>

PetscErrorCode u_exact(DM da, Vec uexact)
{
  DMDALocalInfo info;
  int i, j;
  double hx, hy, x, y, **auexact;

  DMDAGetLocalInfo(da, &info);
  DMDAVecGetArray(da, uexact, &auexact);

  hx = 1.0/(info.mx-1); hy = 1.0/(info.my-1);
  for (j = info.ys; j<(info.ys + info.ym); j++)
  {
    y = j*hy;
    for (i = info.xs; i<(info.xs + info.xm); i++)
    {
      x = i*hx;
      auexact[j][i] = (x*x-x*x*x*x)*(y*y*y*y-y*y);
    }
  }

  DMDAVecRestoreArray(da, uexact, &auexact);
  return 0;
}
```

C:\Users\Admin\Dropbox\TSC_petsc\poisson.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
DMCreateGlobalVector
petsc-3.14.3 2021-01-09
Report Typos and Errors
roy(), DMView(), DMCreateInterpolation(),
DMCreateColoring(), DMCreateMatrix()
C source file
Type here to search

Because we are passing the address of info we can use this kind of a link ok to address the content of the structure ok. So, now we are essentially looping over all the grid points.

(Refer Slide Time: 18:24)



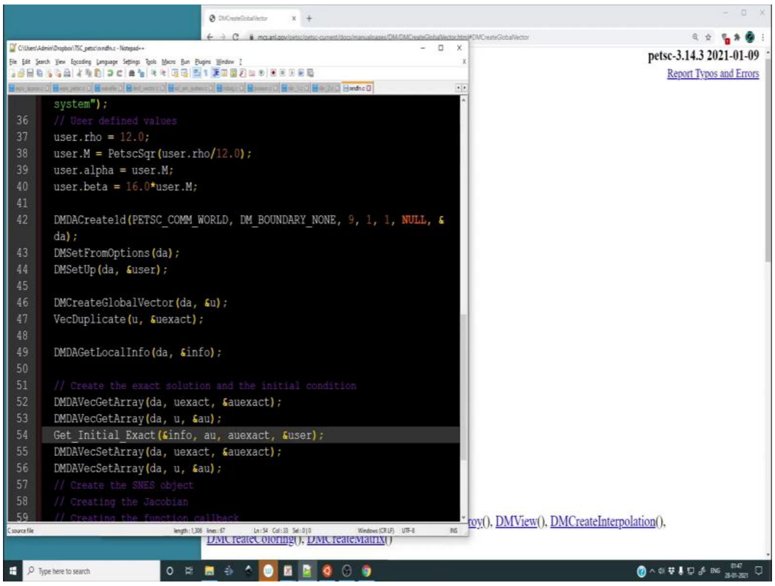
```c
typdef struct
{
  double rho, M, alpha, beta;
} AppCtx;

PetscErrorCode Get_Initial_Exact(DMDALocalInfo *info, double *au,
double *auexact, AppCtx *user)
{
  int i;
  double h = 1.0/(info->mx-1);
  double x;

  for (i = info->xs; i < info->xs+info.xm; i++) // Loop over gridpoints
  {
    x = i*h;
    au[i] = user->alpha*(1.0-x) + user->beta*x;
  }
}



int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
```

petsc-3.14.3 2021-01-09
Report Typos and Errors
roy(), DMView(), DMCreateInterpolation(),
DMCreateColoring(), DMCreateMatrix()

So, this is just a loop over grid points alright. So, inside this loop x the value of x = i*h that is fine and au[i] = user->alpha*(1.0-x) + user->beta*x. So, it is just an initial guess so, we are getting the initial guess inside this u array.

So, look, what is going on. We have the au array which we have passed from main alright, we have get vec array we are getting it we are passing it we are getting the initial condition inside this and then, we will put it back inside u. So finally, we need to so we need to eventually put it back.

(Refer Slide Time: 19:28)

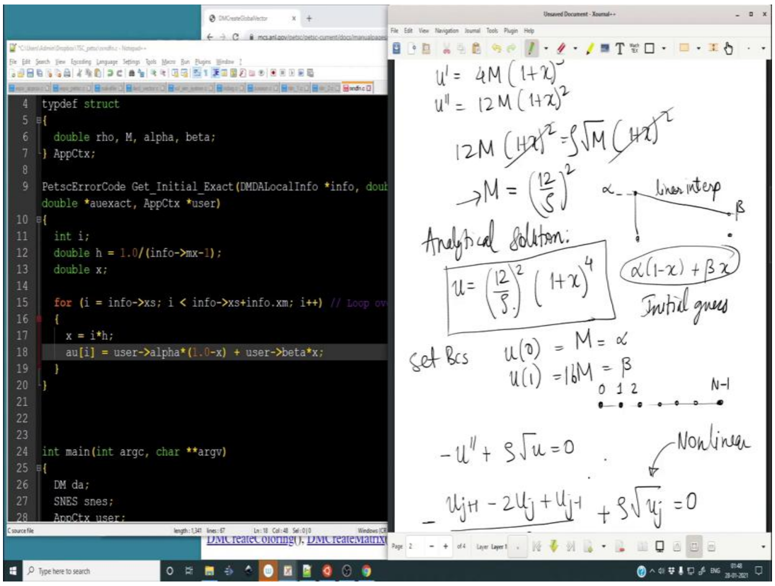


```c
  system");
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.rho/12.0);
  user.alpha = user.M;
  user.beta = 16.0*user.M;

  DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &
  da);
  DMSetFromOptions(da);
  DMSetUp(da, &user);

  DMCreateGlobalVector(da, &u);
  VecDuplicate(u, &uexact);

  DMDAGetLocalInfo(da, &info);

  // Create the exact solution and the initial condition
  DMDAVecGetArray(da, uexact, &auexact);
  DMDAVecGetArray(da, u, &au);
  Get_Initial_Exact(&info, au, auexact, &user);
  DMDAVecSetArray(da, uexact, &auexact);
  DMDAVecSetArray(da, u, &au);
  // Create the SNES object
  // Creating the Jacobian
  // Creating the function callback
```

petsc-3.14.3 2021-01-09
Report Typos and Errors
roy(), DMView(), DMCreateInterpolation(),
DMCreateColoring(), DMCreateMatrix()

So, DMDA SetVecSetArray da uexact & u auexact and we need to copy this, this will be u and this will be au. So, we are fetching the initial conditions through this and then setting it back to the petsc vectors alright. So, au[i] will be equal to it is the guess. So, user->alpha*(1.0-x) + user->beta*x so, what it is? It is a linear interpolation between the end points.

(Refer Slide Time: 20:24)



```
typdef struct
{
  double rho, M, alpha, beta;
} AppCtx;

PetscErrorCode Get_Initial_Exact(DMDALocalInfo *info, doub
double *auexact, AppCtx *user)
{
  int i;
  double h = 1.0/(info->mx-1);
  double x;

  for (i = info->xs; i < info->xs+info.xm; i++) // Loop ov
  {
    x = i*h;
    au[i] = user->alpha*(1.0-x) + user->beta*x;
  }
}



int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
```

$u' = 4M(1+x)^3$

$u'' = 12M(1+x)^2$

$12M\cancel{(1+x)^2} = \rho\sqrt{M}\cancel{(1+x)^2}$

$\rightarrow M = \left(\frac{12}{\rho}\right)^2$

Analytical Solution:

$$\boxed{u = \left(\frac{12}{\rho}\right)^2 (1+x)^4}$$

α linear interp β

$\alpha(1-x) + \beta x$

Initial guess

Set Bcs $u(0) = M = \alpha$

$u(1) = 16M = \beta$

0 1 2 N-1

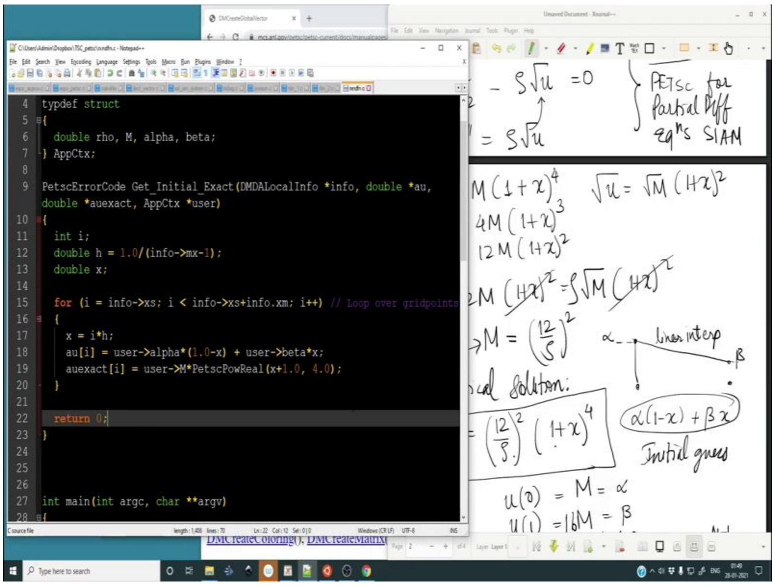
$-u'' + \rho\sqrt{u} = 0$

Nonlinear

$-\frac{u_{j+1} - 2u_j + u_{j-1}}{} + \rho\sqrt{u_j} = 0$

So, these are the two endpoints and I am saying its alpha*(1.0-x)+ beta*x. So, 1 x is 0, the value of the guess is equal to α when the value is equal to 1 this becomes 0 so, the value is β ok. So, its like an initial sorry its like a linear interpolation. So this is the initial guess alright. So, that is the initial guess no problem. Now, we need to set the exact solution.

(Refer Slide Time: 21:04)

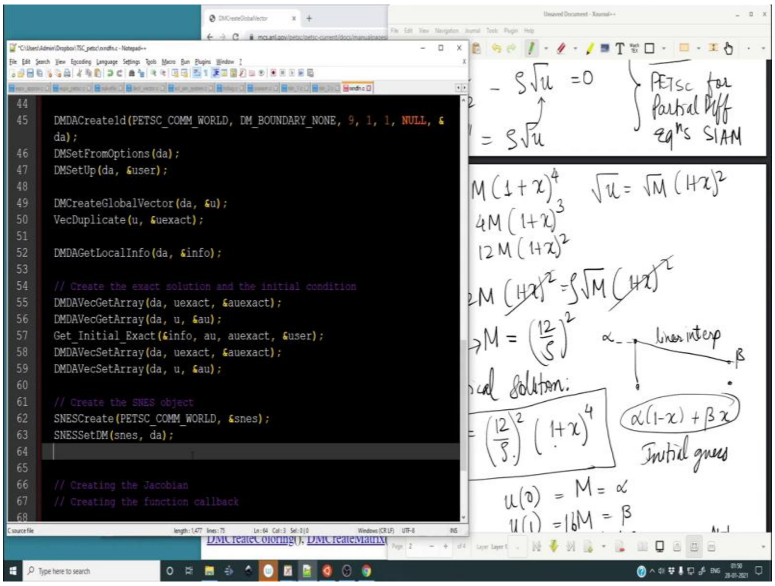


```c
typdef struct
{
  double rho, M, alpha, beta;
} AppCtx;

PetscErrorCode Get_Initial_Exact(DMDALocalInfo *info, double *au,
double *auexact, AppCtx *user)
{
  int i;
  double h = 1.0/(info->mx-1);
  double x;

  for (i = info->xs; i < info->xs+info.xm; i++) // Loop over gridpoints
  {
    x = i*h;
    au[i] = user->alpha*(1.0-x) + user->beta*x;
    auexact[i] = user->M*PetscPowReal(x+1.0, 4.0);
  }

  return 0;
}



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

auexact[i] = user->M*PetscPowReal(x+1.0, 4.0) and this is the not x+ what (Refer Time: 21:28) yeah 1 + x ok. So, this is the solution that we have over here ok.

So, that is the exact solution finally, at the end of this we must return 0 and that is all well and good. So, this is the function which creates the initial condition that we have over here. Finally, we can move on to the creation of this SNES object.

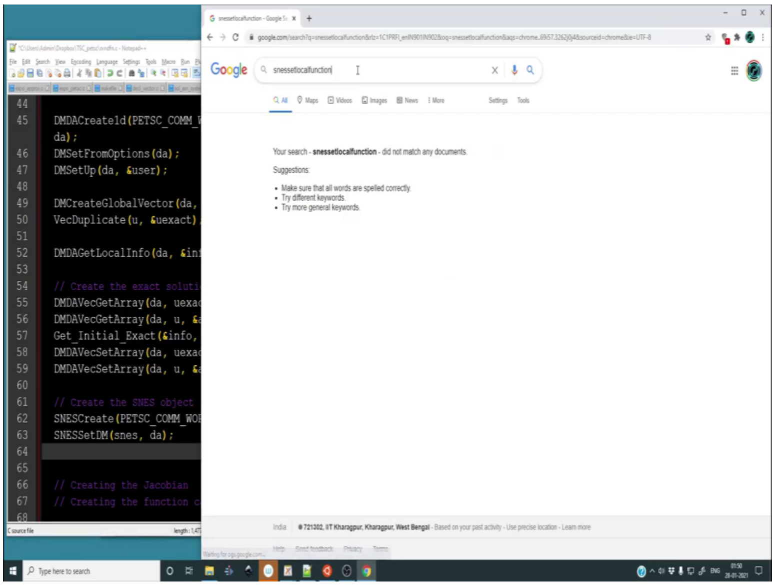
(Refer Slide Time: 21:59)



```c

  DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &
  da);
  DMSetFromOptions(da);
  DMSetUp(da, &user);

  DMCreateGlobalVector(da, &u);
  VecDuplicate(u, &uexact);

  DMDAGetLocalInfo(da, &info);

  // Create the exact solution and the initial condition
  DMDAVecGetArray(da, uexact, &auexact);
  DMDAVecGetArray(da, u, &au);
  Get_Initial_Exact(&info, au, auexact, &user);
  DMDAVecSetArray(da, uexact, &auexact);
  DMDAVecSetArray(da, u, &au);

  // Create the SNES object
  SNESCreate(PETSC_COMM_WORLD, &snes);
  SNESSetDM(snes, da);


  // Creating the Jacobian
  // Creating the function callback
```

So, it is the same protocol SNESCreate PETSC COMM WORLD comma the address of snes right, SNESSetDM so we have to tell that this particular SNES is linked to the distributed array. So, snes sorry snes, da now, we must set the function call alright so we must set here the function callback and the Jacobian alright. So, the function has to be done locally because doing it on a grid ok. So, the way to do it is ok.

(Refer Slide Time: 22:52)



```
DMDACreate1d(PETSC_COMM_W
da);
DMSetFromOptions(da);
DMSetUp(da, &user);

DMCreateGlobalVector(da,
VecDuplicate(u, &uexact)

DMDAGetLocalInfo(da, &inf

// Create the exact soluti
DMDAVecGetArray(da, uexac
DMDAVecGetArray(da, u, &a
Get_Initial_Exact(&info,
DMDAVecSetArray(da, uexac
DMDAVecSetArray(da, u, &a

// Create the SNES object
SNESCreate(PETSC_COMM_WOR
SNESSetDM(snes, da);


// Creating the Jacobian
// Creating the function c
```

snessetlocalfunction

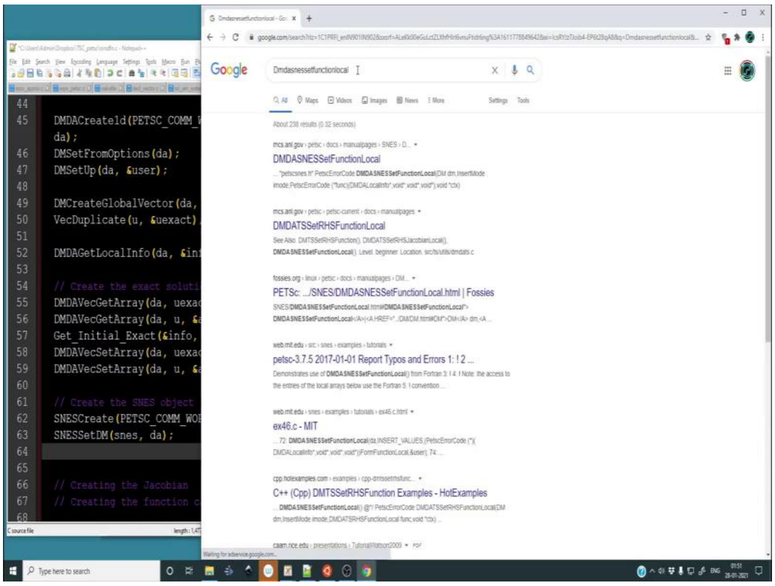
Your search - **snessetlocalfunction** - did not match any documents.

Suggestions:

- Make sure that all words are spelled correctly.
- Try different keywords.
- Try more general keywords.

Let me show you that functional reference snes set local function what is it. Dmda snes set local function local alright.

(Refer Slide Time: 23:02)



```
DMDACreate1d(PETSC_COMM_W
da);
DMSetFromOptions(da);
DMSetUp(da, &user);

DMCreateGlobalVector(da,
VecDuplicate(u, &uexact)

DMDAGetLocalInfo(da, &inf

// Create the exact soluti
DMDAVecGetArray(da, uexac
DMDAVecGetArray(da, u, &a
Get_Initial_Exact(&info,
DMDAVecSetArray(da, uexac
DMDAVecSetArray(da, u, &a

// Create the SNES object
SNESCreate(PETSC_COMM_WOR
SNESSetDM(snes, da);


// Creating the Jacobian
// Creating the function c
```

Dmdasnessetfunctionlocal

About 238 results (0.32 seconds)

mcs.anl.gov › petsc › docs › manualpages › SNES › D...
DMDASNESSetFunctionLocal
... "petscsnes.h" PetscErrorCode **DMDASNESSetFunctionLocal**(DM dm,InsertMode imode,PetscErrorCode (*func)(DMDALocalInfo*,void*,void*,void*),void *ctx)

mcs.anl.gov › petsc › petsc-current › docs › manualpages
DMDATSSetRHSFunctionLocal
See Also. DMTSSetRHSFunction(), DMDATSSetRHSJacobianLocal(), **DMDASNESSetFunctionLocal**(). Level. beginner. Location. src/ts/utils/dmdats.c

fossies.org › linux › petsc › docs › manualpages › DM...
PETSc: .../SNES/DMDASNESSetFunctionLocal.html | Fossies
SNES/**DMDASNESSetFunctionLocal**.html#**DMDASNESSetFunctionLocal**"> **DMDASNESSetFunctionLocal**</A>(<A HREF="../DM/DM.html#DM">DM</A> dm,<A ...

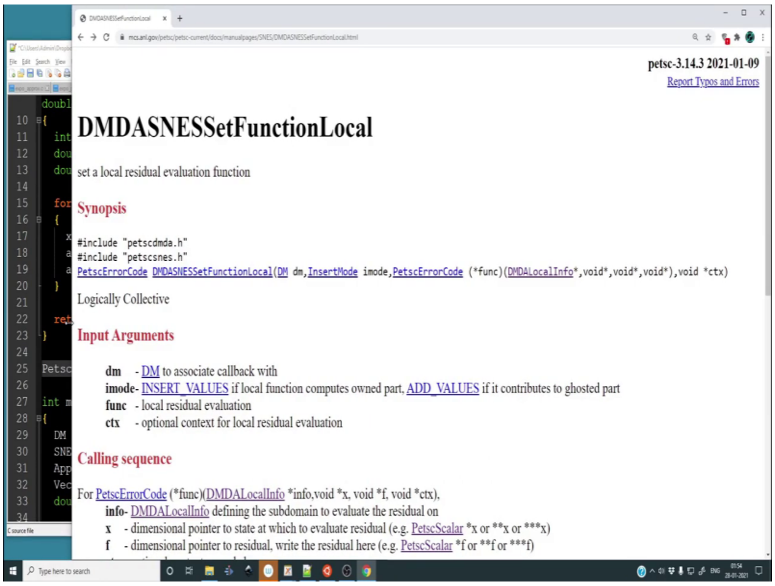
web.mit.edu › src › snes › examples › tutorials
petsc-3.7.5 2017-01-01 Report Typos and Errors 1: ! 2 ...
Demonstrates use of **DMDASNESSetFunctionLocal**() from Fortran 3: ! 4: ! Note: the access to the entries of the local arrays below use the Fortran 5: ! convention ...

web.mit.edu › snes › examples › tutorials › ex46.c.html
ex46.c - MIT
... 72: **DMDASNESSetFunctionLocal**(da,INSERT_VALUES,(PetscErrorCode (*)(DMDALocalInfo*,void*,void*,void*))FormFunctionLocal,&user); 74: ...

cpp.hotexamples.com › examples › cpp-dmtssetrhsfunc...
C++ (Cpp) DMTSSetRHSFunction Examples - HotExamples
... **DMDASNESSetFunctionLocal**() @*/ PetscErrorCode DMDATSSetRHSFunctionLocal(DM dm,InsertMode imode,DMDATSRHSFunctionLocal func,void *ctx) ...

caam.rice.edu › presentations › TutorialWatson2009

(Refer Slide Time: 23:14)



```
doubl
{
  int
  dou
  dou

  for
  {
    x
    a
    a
  }

  ret
}

Petsc

int m
{
  DM
  SNE
  App
  Vec
  dou
```

petsc-3.14.3 2021-01-09
Report Typos and Errors

DMDASNESSetFunctionLocal

set a local residual evaluation function

Synopsis

```
#include "petscdmda.h"
#include "petscsnes.h"
PetscErrorCode DMDASNESSetFunctionLocal(DM dm,InsertMode imode,PetscErrorCode (*func)(DMDALocalInfo*,void*,void*,void*),void *ctx)
```

Logically Collective

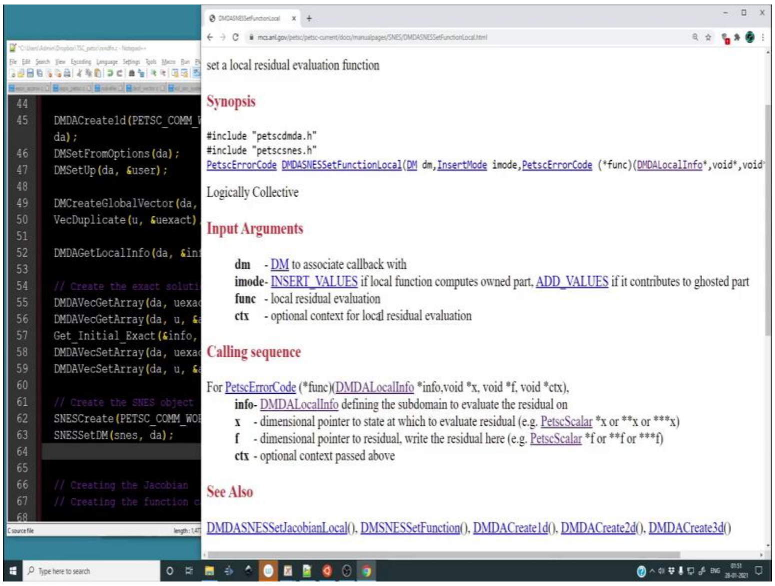
Input Arguments

dm - DM to associate callback with
imode- INSERT_VALUES if local function computes owned part, ADD_VALUES if it contributes to ghosted part
func - local residual evaluation
ctx - optional context for local residual evaluation

Calling sequence

For PetscErrorCode (*func)(DMDALocalInfo *info,void *x, void *f, void *ctx),
info- DMDALocalInfo defining the subdomain to evaluate the residual on
x - dimensional pointer to state at which to evaluate residual (e.g. PetscScalar *x or **x or ***x)
f - dimensional pointer to residual, write the residual here (e.g. PetscScalar *f or **f or ***f)

(Refer Slide Time: 23:21)



```
DMDACreate1d(PETSC_COMM_W
da);
DMSetFromOptions(da);
DMSetUp(da, &user);

DMCreateGlobalVector(da,
VecDuplicate(u, &uexact)

DMDAGetLocalInfo(da, &in

// Create the exact soluti
DMDAVecGetArray(da, uexac
DMDAVecGetArray(da, u, &
Get_Initial_Exact(&info,
DMDAVecSetArray(da, uexac
DMDAVecSetArray(da, u, &

// Create the SNES object
SNESCreate(PETSC_COMM_WOR
SNESSetDM(snes, da);


// Creating the Jacobian
// Creating the function c
```

set a local residual evaluation function

Synopsis

```
#include "petscdmda.h"
#include "petscsnes.h"
PetscErrorCode DMDASNESSetFunctionLocal(DM dm,InsertMode imode,PetscErrorCode (*func)(DMDALocalInfo*,void*,void
```

Logically Collective

Input Arguments

dm - DM to associate callback with
imode- INSERT_VALUES if local function computes owned part, ADD_VALUES if it contributes to ghosted part
func - local residual evaluation
ctx - optional context for local residual evaluation

Calling sequence

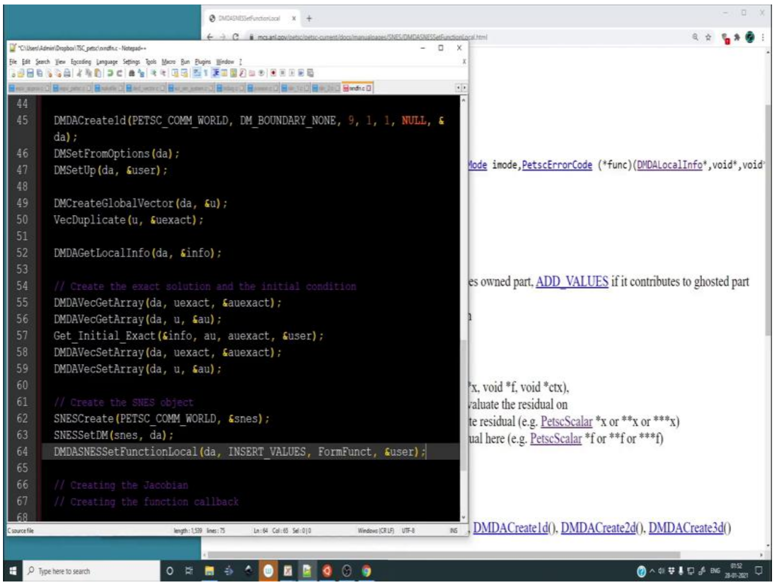
For PetscErrorCode (*func)(DMDALocalInfo *info,void *x, void *f, void *ctx),
info- DMDALocalInfo defining the subdomain to evaluate the residual on
x - dimensional pointer to state at which to evaluate residual (e.g. PetscScalar *x or **x or ***x)
f - dimensional pointer to residual, write the residual here (e.g. PetscScalar *f or **f or ***f)
ctx - optional context passed above

See Also

DMDASNESSetJacobianLocal(), DMSNESSetFunction(), DMDACreate1d(), DMDACreate2d(), DMDACreate3d()

So, it is a residual evaluation over a DM ok. So, because the function callback in this particular problem is in the form of a matrix which is assembled over the distr and the dmda ok so, that is why we need to do the following.

(Refer Slide Time: 23:41)



```
DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &
da);
DMSetFromOptions(da);
DMSetUp(da, &user);

DMCreateGlobalVector(da, &u);
VecDuplicate(u, &uexact);

DMDAGetLocalInfo(da, &info);

// Create the exact solution and the initial condition
DMDAVecGetArray(da, uexact, &auexact);
DMDAVecGetArray(da, u, &au);
Get_Initial_Exact(&info, au, auexact, &user);
DMDAVecSetArray(da, uexact, &auexact);
DMDAVecSetArray(da, u, &au);

// Create the SNES object
SNESCreate(PETSC_COMM_WORLD, &snes);
SNESSetDM(snes, da);
DMDASNESSetFunctionLocal(da, INSERT_VALUES, FormFunct, &user);

// Creating the Jacobian
// Creating the function callback
```

Mode imode,PetscErrorCode (*func)(DMDALocalInfo*,void*,void*

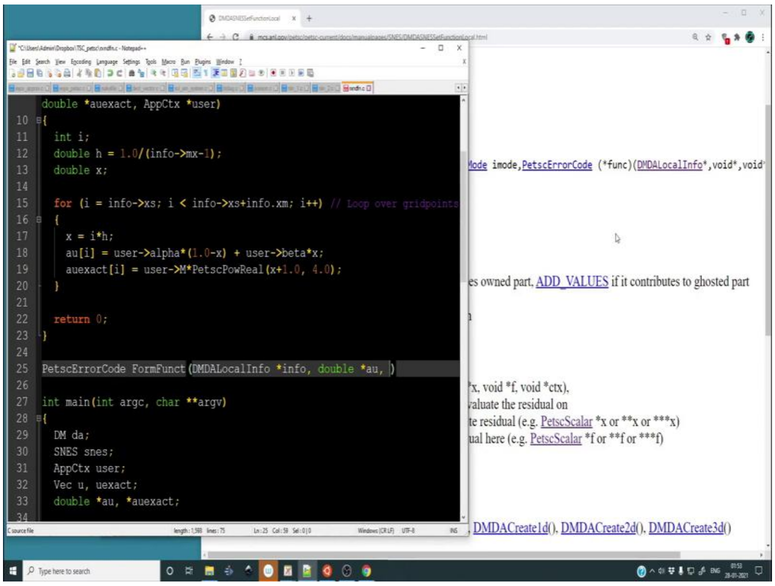
es owned part, ADD_VALUES if it contributes to ghosted part

*x, void *f, void *ctx),
valuate the residual on
te residual (e.g. PetscScalar *x or **x or ***x)
ual here (e.g. PetscScalar *f or **f or ***f)

, DMDACreate1d(), DMDACreate2d(), DMDACreate3d()

So, we have DMDASNESSetFunctionLocal over da and we must do INSERT VALUES. So, that is the way it is dm mode function and ctx. So DM mode finally, we need to define the function as well. So, the function will be FormFunct and the user context. So now, let us go ahead and define this function local ok.

(Refer Slide Time: 24:29)



```
double *auexact, AppCtx *user)
{
  int i;
  double h = 1.0/(info->mx-1);
  double x;

  for (i = info->xs; i < info->xs+info.xm; i++) // Loop over gridpoints
  {
    x = i*h;
    au[i] = user->alpha*(1.0-x) + user->beta*x;
    auexact[i] = user->M*PetscPowReal(x+1.0, 4.0);
  }

  return 0;
}

PetscErrorCode FormFunct(DMDALocalInfo *info, double *au, )

int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;
  double *au, *auexact;
```

Mode imode,PetscErrorCode (*func)(DMDALocalInfo*,void*,void*

es owned part, ADD_VALUES if it contributes to ghosted part

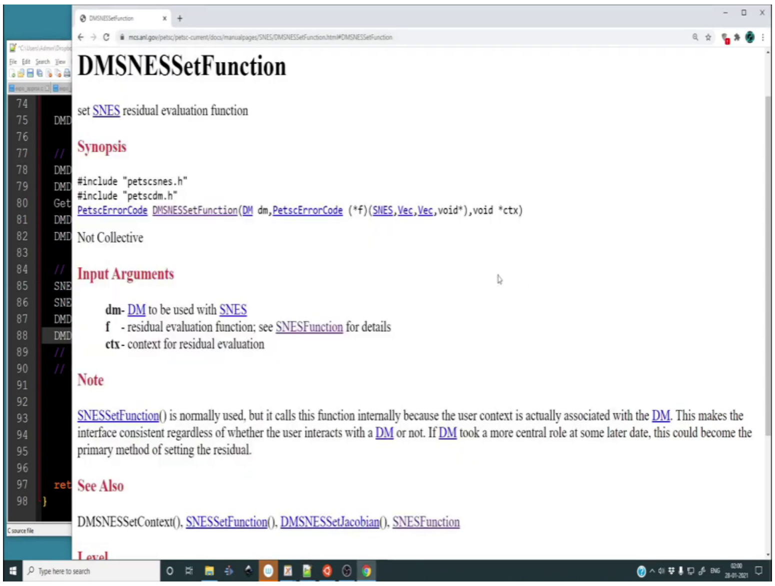
*x, void *f, void *ctx),
valuate the residual on
te residual (e.g. PetscScalar *x or **x or ***x)
ual here (e.g. PetscScalar *f or **f or ***f)

, DMDACreate1d(), DMDACreate2d(), DMDACreate3d()

So, PetscErrorCode what was so, it is a pointer to the function ok let is see if there is an error we can always fix it ok. So, PetscErrorCodeFormFunct and DMDA so we will have the same type of inputs so, DMDALocalInfo star info what else do we have, we have the array so double *u in this case it is au ok.

So it is actually a local function evaluation and we have to pass the handle to that ok. So, we will rectify it later. So, the function should take as an input the info because we need to perform a loop then dmdas.

(Refer Slide Time: 25:55)



DMSNESSetFunction

set SNES residual evaluation function

Synopsis

```
#include "petscsnes.h"
#include "petscdm.h"
PetscErrorCode DMSNESSetFunction(DM dm,PetscErrorCode (*f)(SNES,Vec,Vec,void*),void *ctx)
```

Not Collective

Input Arguments

dm- DM to be used with SNES
f - residual evaluation function; see SNESFunction for details
ctx- context for residual evaluation

Note

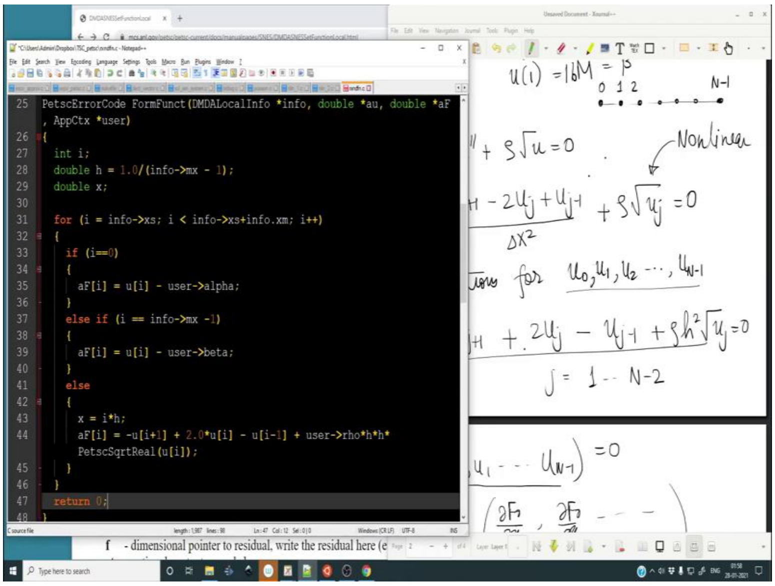
SNESSetFunction() is normally used, but it calls this function internally because the user context is actually associated with the DM. This makes the interface consistent regardless of whether the user interacts with a DM or not. If DM took a more central role at some later date, this could become the primary method of setting the residual.

See Also

DMSNESSetContext(), SNESSetFunction(), DMSNESSetJacobian(), SNESFunction

So, we need to wait ok so, it will take the u array and the evaluation of the function.

(Refer Slide Time: 26:36)



```
PetscErrorCode FormFunct(DMDALocalInfo *info, double *au, double *aF
, AppCtx *user)
{
  int i;
  double h = 1.0/(info->mx - 1);
  double x;

  for (i = info->xs; i < info->xs+info.xm; i++)
  {
    if (i==0)
    {
      aF[i] = u[i] - user->alpha;
    }
    else if (i == info->mx -1)
    {
      aF[i] = u[i] - user->beta;
    }
    else
    {
      x = i*h;
      aF[i] = -u[i+1] + 2.0*u[i] - u[i-1] + user->rho*h*h*
      PetscSqrtReal(u[i]);
    }
  }
  return 0;
}
```

f - dimensional pointer to residual, write the residual here (e

$u(1) = 16M = \beta$
0 1 2 N-1
$'' + \rho\sqrt{u} = 0$
Nonlinear
$\frac{{}_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} + \rho\sqrt{u_j} = 0$
for $u_0, u_1, u_2 \cdots, u_{N-1}$
${}_{j+1} + 2u_j - u_{j-1} + \rho h^2\sqrt{u_j} = 0$
$j = 1 \cdots N-2$
$, u_1 \cdots u_{N-1}) = 0$
$\left(\frac{\partial F_1}{\partial}, \frac{\partial F_1}{\partial} \cdots\right)$

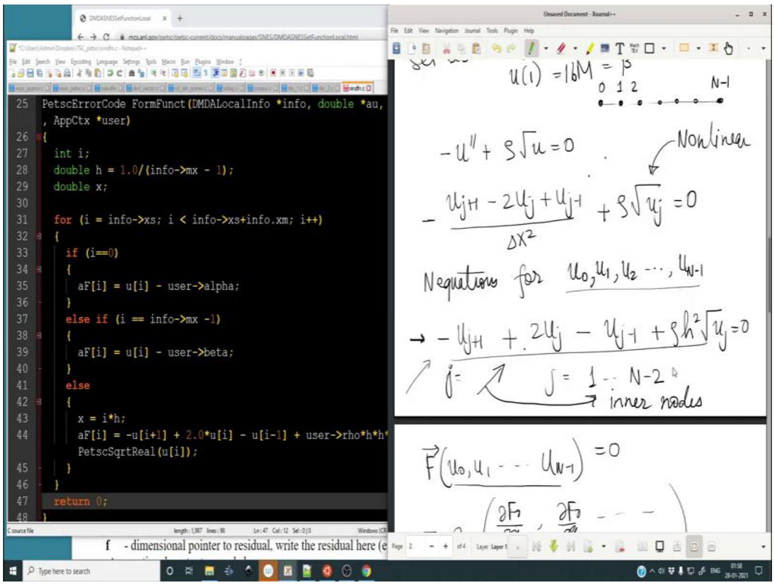
So, it will be double *aF and the ApplicationContext *user. So, what we need to do over here is to create the functions in the particular problem. So the functions will be where is it functions will be these things ok. So, and those will be stored in ff aF rather ok. So, int i const double because we do not want to change h this will be simply we can just make it double we are not going to change it anyway (info->mx - 1) alright.

So, we need double to store x and the reaction term or we can just hold it in x, we can define it later on we will then input these things these particular things ok. So double x what we need to do for. So, we need this same for loop to loop over all the nodes that is it then, we have if (i==0) then aF[i] = au[i] - user->alpha; else if (i == info->mx -1) that is the right hand side grid or the rather right hand side node then, aF[i] = au[i] - user->beta alright else you simply have the interior points. So, the interior points are easy.

So, what is the function? So, because it is happening locally that is why so first we need to have x = i*h, which we have already defined and yep. So, aF[i] like the ith function will be in terms of the u which is on the local grid.

So, -u[i+1] + 2.0*u[i] - u[i-1] + user->rho*h*h*PetscSqrtReal(u[i]) so, that is the way we will declare the function and once it is successful we will simply return 0 ok.

(Refer Slide Time: 30:38)



```
PetscErrorCode FormFunct(DMDALocalInfo *info, double *au,
, AppCtx *user)
{
  int i;
  double h = 1.0/(info->mx - 1);
  double x;

  for (i = info->xs; i < info->xs+info.xm; i++)
  {
    if (i==0)
    {
      aF[i] = u[i] - user->alpha;
    }
    else if (i == info->mx -1)
    {
      aF[i] = u[i] - user->beta;
    }
    else
    {
      x = i*h;
      aF[i] = -u[i+1] + 2.0*u[i] - u[i-1] + user->rho*h*h*
      PetscSqrtReal(u[i]);
    }
  }
  return 0;
}
```

f - dimensional pointer to residual, write the residual here (e

$u(1) = 16M = \beta$

0 1 2 N-1

$-u'' + \rho\sqrt{u} = 0$ Nonlinear

$-\frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} + \rho\sqrt{u_j} = 0$

N equations for $u_0, u_1, u_2 \dots, u_{N-1}$

$\rightarrow -u_{j+1} + 2u_j - u_{j-1} + \rho h^2\sqrt{u_j} = 0$

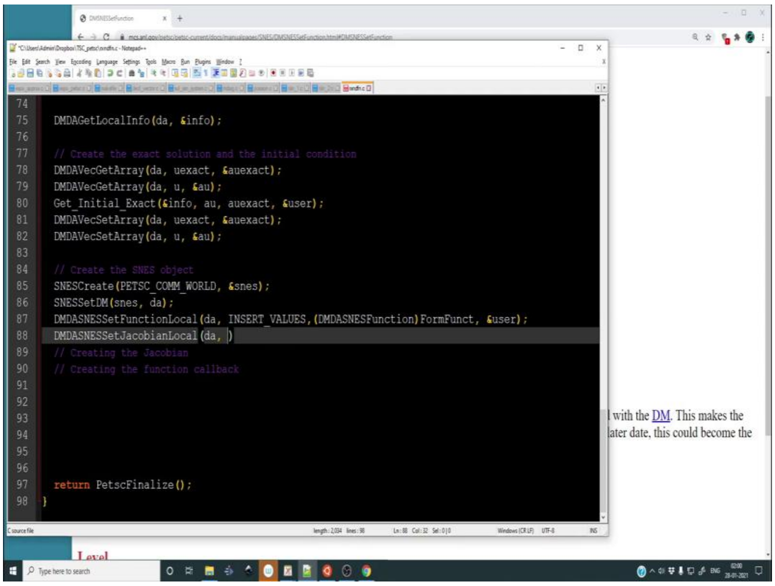
$j =$ $j = 1 \dots N-2$ inner nodes

$\vec{F}(u_0, u_1 \dots u_{N-1}) = 0$

$\left(\frac{\partial F_0}{\partial u_0}, \frac{\partial F_0}{\partial u_1} \dots\right)$

So, we have gone ahead and created this function. So, all these functions are valid for j equal to the inner node so, these are all the inner nodes. Because the residues will be calculated for these function evaluations ok. So, we have DMDAsSetLocalFunctionSetFunctionLocal and it requires what let us see da then the PetscErrorCode no not the PetscError the InsertMode then the function handle to so, it needs the function handle.

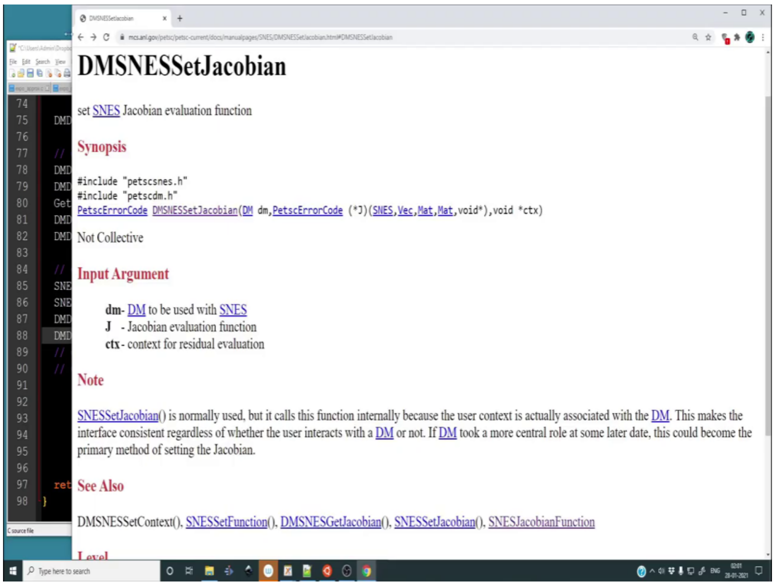
(Refer Slide Time: 31:33)



```
  DMDAGetLocalInfo(da, &info);

  // Create the exact solution and the initial condition
  DMDAVecGetArray(da, uexact, &auexact);
  DMDAVecGetArray(da, u, &au);
  Get_Initial_Exact(&info, au, auexact, &user);
  DMDAVecSetArray(da, uexact, &auexact);
  DMDAVecSetArray(da, u, &au);

  // Create the SNES object
  SNESCreate(PETSC_COMM_WORLD, &snes);
  SNESSetDM(snes, da);
  DMDASNESSetFunctionLocal(da, INSERT_VALUES,(DMDASNESFunction)FormFunct, &user);
  DMDASNESSetJacobianLocal(da, )
  // Creating the Jacobian
  // Creating the function callback






  return PetscFinalize();
}
```

So, we need to give instead of giving the name of the function we need to give the Function handle. So, we will say that form funct is of the type DMDASNESFunction ok. So, this will make it pass the function handle. It should run without even declaring it like this, but anyway let us see if it does not work then we will think about it.

(Refer Slide Time: 32:38)



DMSNESSetJacobian

set SNES Jacobian evaluation function

Synopsis

```
#include "petscsnes.h"
#include "petscdm.h"
PetscErrorCode DMSNESSetJacobian(DM dm,PetscErrorCode (*J)(SNES,Vec,Mat,Mat,void*),void *ctx)
```

Not Collective

Input Argument

dm- DM to be used with SNES
J - Jacobian evaluation function
ctx- context for residual evaluation

Note

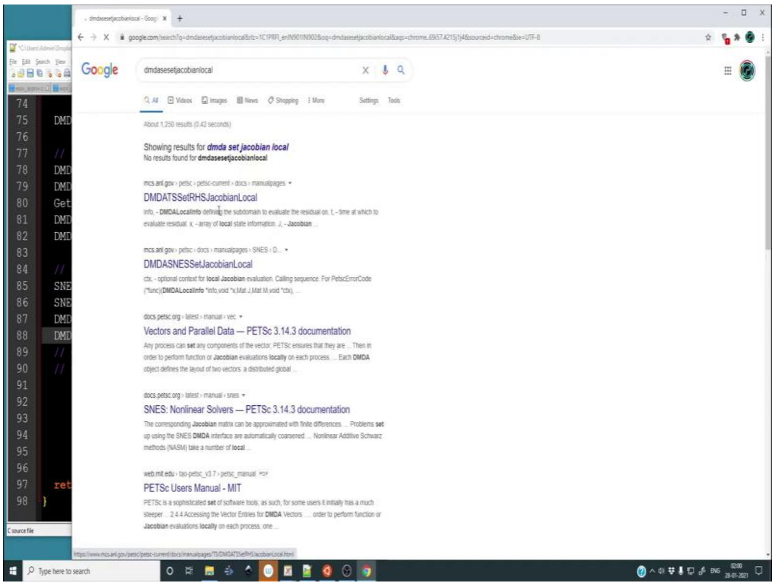
SNESSetJacobian() is normally used, but it calls this function internally because the user context is actually associated with the DM. This makes the interface consistent regardless of whether the user interacts with a DM or not. If DM took a more central role at some later date, this could become the primary method of setting the Jacobian.

See Also

DMSNESSetContext(), SNESSetFunction(), DMSNESGetJacobian(), SNESSetJacobian(), SNESJacobianFunction

Now, let me make the Jacobian callback so, DMDASetSNESSetJacobianLocal alright. So, even this will contain da then the function callback if I remember correctly.

(Refer Slide Time: 32:48)



dmdasesetjacobianlocal

About 1,250 results (0.42 seconds)

Showing results for ***dmda set jacobian local***
No results found for **dmdasesetjacobianlocal**

mcs.anl.gov › petsc › petsc-current › docs › manualpages
DMDATSSetRHSJacobianLocal
info, - **DMDALocalInfo** defining the subdomain to evaluate the residual on. t, - time at which to evaluate residual. x, - array of **local** state information. J, - **Jacobian** ...

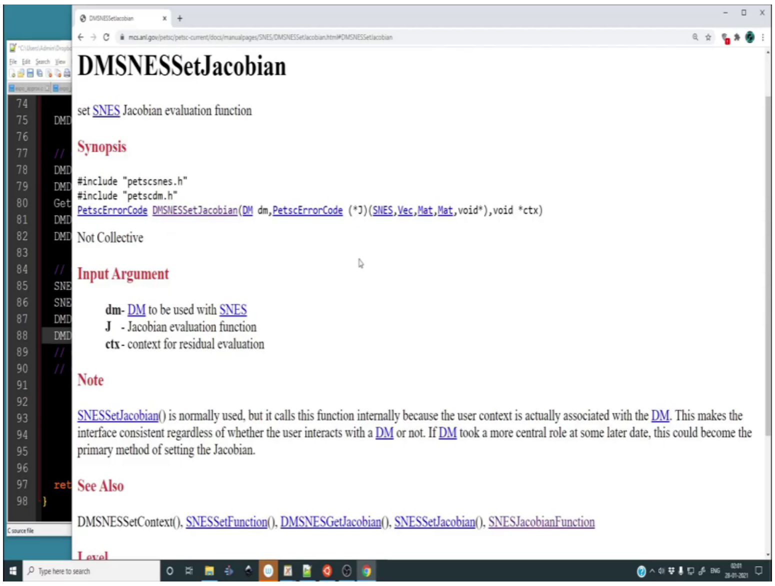
mcs.anl.gov › petsc › docs › manualpages › SNES › D...
DMDASNESSetJacobianLocal
ctx, - optional context for **local Jacobian** evaluation. Calling sequence. For PetscErrorCode (*func)(**DMDALocalInfo** *info,void *x,Mat J,Mat M,void *ctx), ...

docs.petsc.org › latest › manual › vec
Vectors and Parallel Data — PETSc 3.14.3 documentation
Any process can **set** any components of the vector; PETSc ensures that they are ... Then in order to perform function or **Jacobian** evaluations **locally** on each process. ... Each **DMDA** object defines the layout of two vectors: a distributed global ...

docs.petsc.org › latest › manual › snes
SNES: Nonlinear Solvers — PETSc 3.14.3 documentation
The corresponding **Jacobian** matrix can be approximated with finite differences. ... Problems **set** up using the SNES **DMDA** interface are automatically coarsened. ... Nonlinear Additive Schwarz methods (NASM) take a number of **local** ...

web.mit.edu › tao-petsc_v3.7 › petsc_manual PDF
PETSc Users Manual - MIT
PETSc is a sophisticated **set** of software tools; as such, for some users it initially has a much steeper ... 2.4.4 Accessing the Vector Entries for **DMDA** Vectors. ... order to perform function or **Jacobian** evaluations **locally** on each process, one ...

(Refer Slide Time: 32:56)



DMSNESSetJacobian

set SNES Jacobian evaluation function

Synopsis

```
#include "petscsnes.h"
#include "petscdm.h"
PetscErrorCode DMSNESSetJacobian(DM dm,PetscErrorCode (*J)(SNES,Vec,Mat,Mat,void*),void *ctx)
```

Not Collective

Input Argument

- **dm**- DM to be used with SNES
- **J** - Jacobian evaluation function
- **ctx**- context for residual evaluation

Note

SNESSetJacobian() is normally used, but it calls this function internally because the user context is actually associated with the DM. This makes the interface consistent regardless of whether the user interacts with a DM or not. If DM took a more central role at some later date, this could become the primary method of setting the Jacobian.

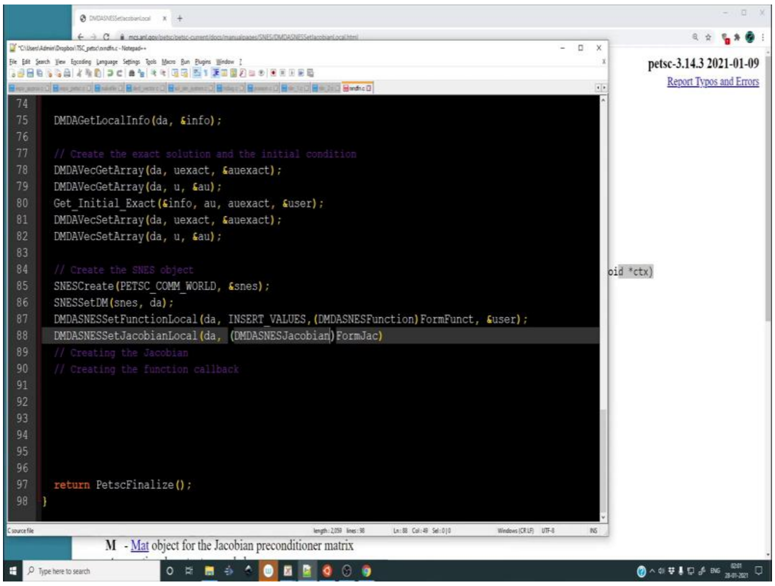
See Also

DMSNESSetContext(), SNESSetFunction(), DMSNESGetJacobian(), SNESSetJacobian(), SNESJacobianFunction

Level

So, DMDAS as JacobianLocal DMDASNESSetJacobianLocal it is the same kind of function handle. So, dm then the function handle and the matrices ok.

(Refer Slide Time: 33:19)



```
    DMDAGetLocalInfo(da, &info);

    // Create the exact solution and the initial condition
    DMDAVecGetArray(da, uexact, &auexact);
    DMDAVecGetArray(da, u, &au);
    Get_Initial_Exact(&info, au, auexact, &user);
    DMDAVecSetArray(da, uexact, &auexact);
    DMDAVecSetArray(da, u, &au);

    // Create the SNES object
    SNESCreate(PETSC_COMM_WORLD, &snes);
    SNESSetDM(snes, da);
    DMDASNESSetFunctionLocal(da, INSERT_VALUES,(DMDASNESFunction)FormFunct, &user);
    DMDASNESSetJacobianLocal(da, (DMDASNESJacobian)FormJac)
    // Creating the Jacobian
    // Creating the function callback






    return PetscFinalize();
}
```

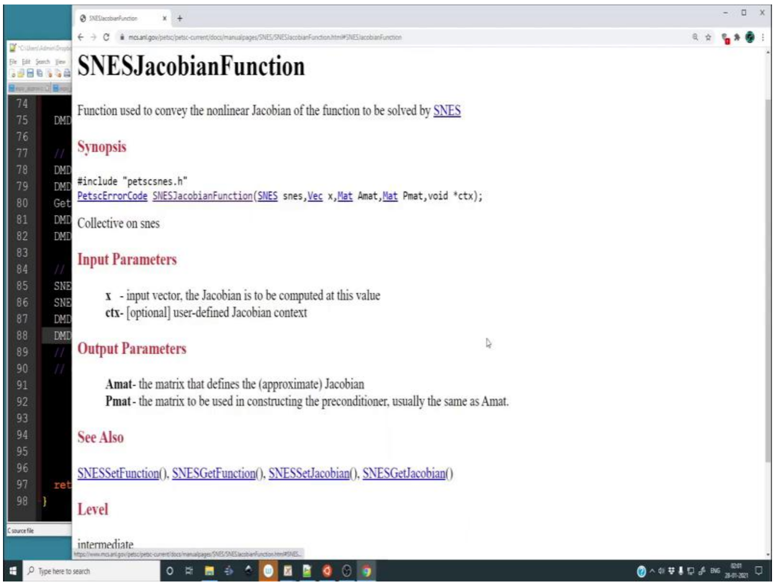
petsc-3.14.3 2021-01-09
Report Typos and Errors

oid *ctx)

M - Mat object for the Jacobian preconditioner matrix

Actually the function handle should contain the matrices and the application context ok. So, this should contain the function handle so, FormJac and the type of the function is DMDASNES function and not function Jacobian ok.

(Refer Slide Time: 33:45)



SNESJacobianFunction

Function used to convey the nonlinear Jacobian of the function to be solved by SNES

Synopsis

```
#include "petscsnes.h"
PetscErrorCode SNESJacobianFunction(SNES snes,Vec x,Mat Amat,Mat Pmat,void *ctx);
```

Collective on snes

Input Parameters

x - input vector, the Jacobian is to be computed at this value
ctx- [optional] user-defined Jacobian context

Output Parameters

Amat- the matrix that defines the (approximate) Jacobian
Pmat- the matrix to be used in constructing the preconditioner, usually the same as Amat.

See Also

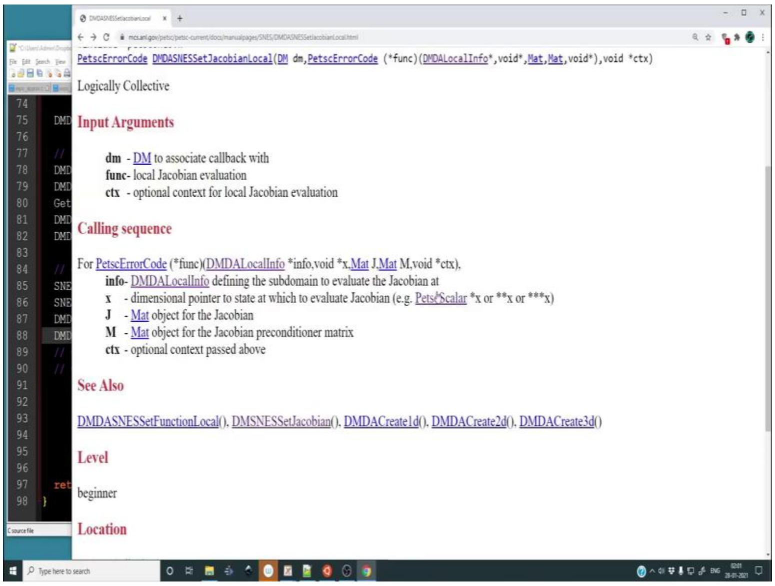
SNESSetFunction(), SNESGetFunction(), SNESSetJacobian(), SNESGetJacobian()

Level

intermediate

It has to give up this of that particular kind because, when you said Jacobian the Jacobian function has to be of this particular kind ok, alright.

(Refer Slide Time: 33:58)



```
PetscErrorCode DMDASNESSetJacobianLocal(DM dm,PetscErrorCode (*func)(DMDALocalInfo*,void*,Mat,Mat,void*),void *ctx)
```

Logically Collective

Input Arguments

dm - DM to associate callback with
func- local Jacobian evaluation
ctx - optional context for local Jacobian evaluation

Calling sequence

For PetscErrorCode (*func)(DMDALocalInfo *info,void *x,Mat J,Mat M,void *ctx),
info- DMDALocalInfo defining the subdomain to evaluate the Jacobian at
x - dimensional pointer to state at which to evaluate Jacobian (e.g. PetscScalar *x or **x or ***x)
J - Mat object for the Jacobian
M - Mat object for the Jacobian preconditioner matrix
ctx - optional context passed above

See Also

DMDASNESSetFunctionLocal(), DMSNESSetJacobian(), DMDACreate1d(), DMDACreate2d(), DMDACreate3d()

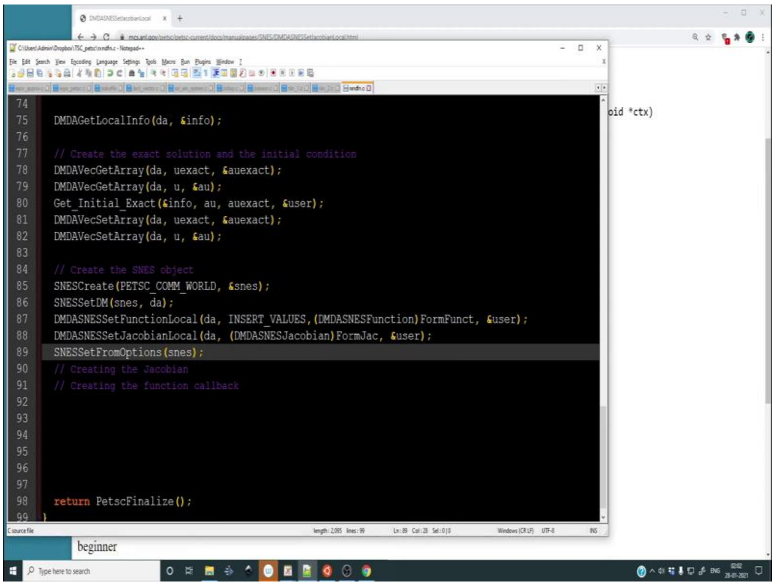
Level

beginner

Location

Then ok so apart from this, we need to pass the application context as well so, the address of user ok.

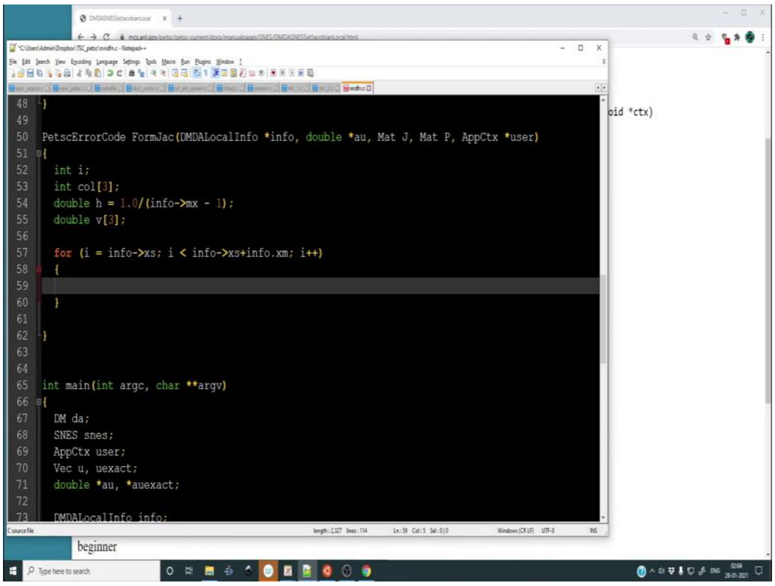
(Refer Slide Time: 34:05)



```
  DMDAGetLocalInfo(da, &info);

  // Create the exact solution and the initial condition
  DMDAVecGetArray(da, uexact, &auexact);
  DMDAVecGetArray(da, u, &au);
  Get_Initial_Exact(&info, au, auexact, &user);
  DMDAVecSetArray(da, uexact, &auexact);
  DMDAVecSetArray(da, u, &au);

  // Create the SNES object
  SNESCreate(PETSC_COMM_WORLD, &snes);
  SNESSetDM(snes, da);
  DMDASNESSetFunctionLocal(da, INSERT_VALUES,(DMDASNESFunction)FormFunct, &user);
  DMDASNESSetJacobianLocal(da, (DMDASNESJacobian)FormJac, &user);
  SNESSetFromOptions(snes);
  // Creating the Jacobian
  // Creating the function callback







  return PetscFinalize();
}
```

So, then we need to create the function, but before that SNESSetFromOptions snes ok. So, now we need to create this form jack function. So, the form is the same as the above it needs one which created from scratch so, it is essentially going to return again an integer to say whether everything has run properly or not.

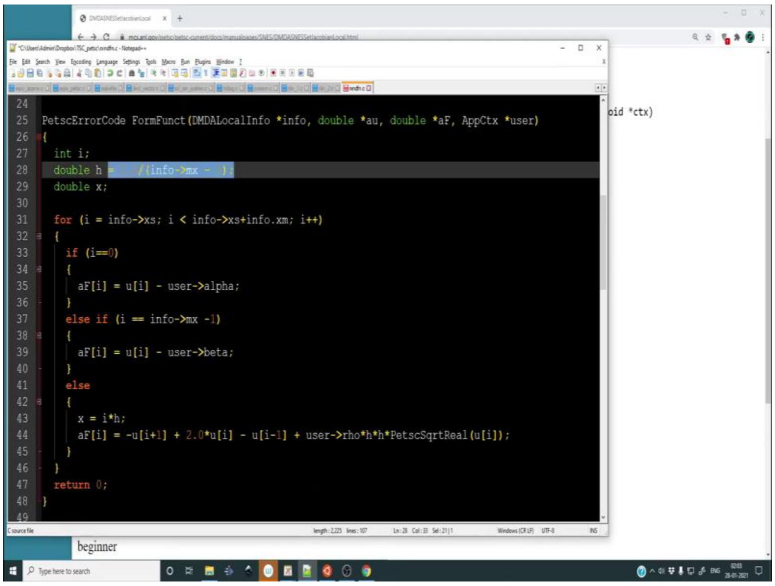
(Refer Slide Time: 34:38)



```
}

PetscErrorCode FormJac(DMDALocalInfo *info, double *au, Mat J, Mat P, AppCtx *user)
{
  int i;
  int col[3];
  double h = 1.0/(info->mx - 1);
  double v[3];

  for (i = info->xs; i < info->xs+info.xm; i++)
  {

  }

}


int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;
  double *au, *auexact;

  DMDALocalInfo info;
```

So, PetscErrorCode FormJac(DMDALocalInfo *info, double *au, Mat J, Mat P, AppCtx *user). It is the same thing, instead of passing the vector we are passing the auxiliary vector that is it that is the only difference ok, alright. So, int i and for the Jacobian we going to have to insert three columns. So, int col[3] and they will keep on shifting depending on i we need a double h, which we can borrow from the previous code alright.

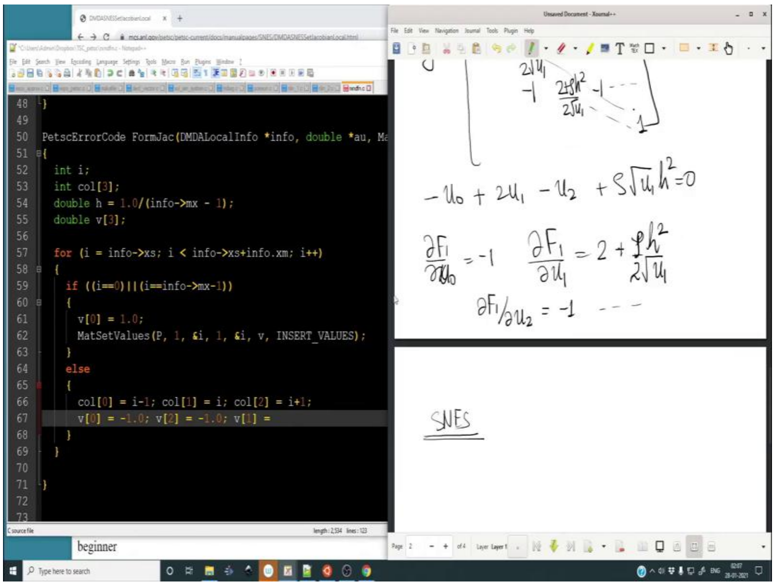
(Refer Slide Time: 36:03)



```c
PetscErrorCode FormFunct(DMDALocalInfo *info, double *au, double *aF, AppCtx *user)
{
  int i;
  double h = 1.0/(info->mx - 1);
  double x;

  for (i = info->xs; i < info->xs+info.xm; i++)
  {
    if (i==0)
    {
      aF[i] = u[i] - user->alpha;
    }
    else if (i == info->mx -1)
    {
      aF[i] = u[i] - user->beta;
    }
    else
    {
      x = i*h;
      aF[i] = -u[i+1] + 2.0*u[i] - u[i-1] + user->rho*h*h*PetscSqrtReal(u[i]);
    }
  }
  return 0;
}
```

And we going to need something which will store 3 values because, we will insert the 3 values ok. So, then we will do the for loop for so, we going to borrow the for loop alright. Now, if i == 0 so, what is the Jacobian we have already discussed this, but when i == 0 and i = n - 1 the Jacobian the diagonal element is simply going to be 1 alright.

(Refer Slide Time: 37:00)



```c
}

PetscErrorCode FormJac(DMDALocalInfo *info, double *au, Ma
{
  int i;
  int col[3];
  double h = 1.0/(info->mx - 1);
  double v[3];

  for (i = info->xs; i < info->xs+info.xm; i++)
  {
    if ((i==0)||(i==info->mx-1))
    {
      v[0] = 1.0;
      MatSetValues(P, 1, &i, 1, &i, v, INSERT_VALUES);
    }
    else
    {
      col[0] = i-1; col[1] = i; col[2] = i+1;
      v[0] = -1.0; v[2] = -1.0; v[1] =
    }
  }

}
```

$2\sqrt{u_1}$

$-1 \quad \frac{2+\rho h^2}{2\sqrt{u_1}} \quad -1 \; \cdots \; 1$

$$-u_0 + 2u_1 - u_2 + \rho\sqrt{u_1}\,h^2 = 0$$

$$\frac{\partial F_1}{\partial u_0} = -1 \qquad \frac{\partial F_1}{\partial u_1} = 2 + \frac{\rho h^2}{2\sqrt{u_1}}$$

$$\partial F_1/\partial u_2 = -1 \quad \cdots$$

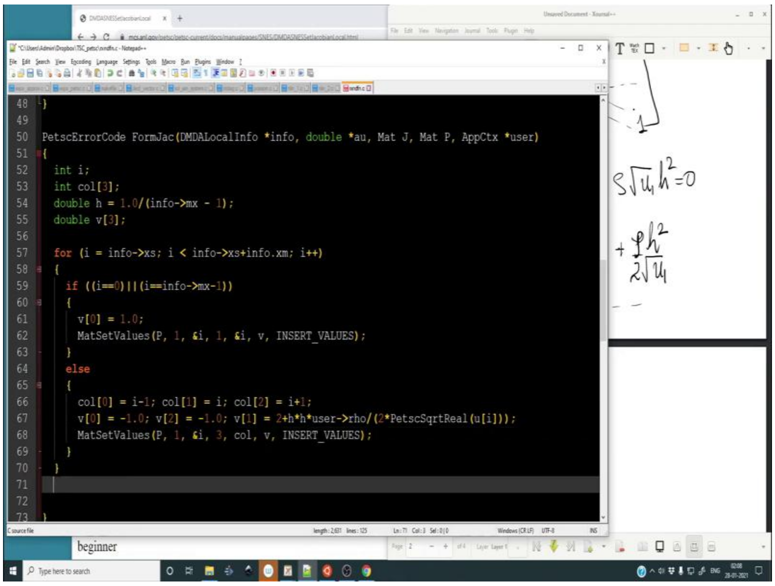
SNES

So, let us encode that so, if i = 0 then what should we do? v[0] = 1.0 and col [0] does not matter we can choose not to give call because it is going to be the same location as i. So, its we going to insert it at i, i. So, we going to do MatSetValues into P and we going to insert 1 row, the location of the row is i and we going to insert 1 column and the location of the column is also i that is why we do not need to bother about row.

And we going to insert v because, we going to insert only one value v [0] is sufficient and we going to do it as insert values. In fact, when i is 0 or 1 i is equal to the last row i==info->mx-1. So, when this is true then this needs to be done ok. So, we have to set the Jacobian to one now, when it is not the case else what should we do else you have to construct the 3 rows and do it ok.

So, col[0] = i-1, col[1] = i alright and col[i+1] is going to be not i+1 col[2] = i+1, these are the columns where we want to insert the values alright. Now, v[0] = -1.0, v[2] = -1.0 whereas, v[1] = 2+h*h*user->rho/(2*PetscSqrtReal(au[i])).

(Refer Slide Time: 39:51)

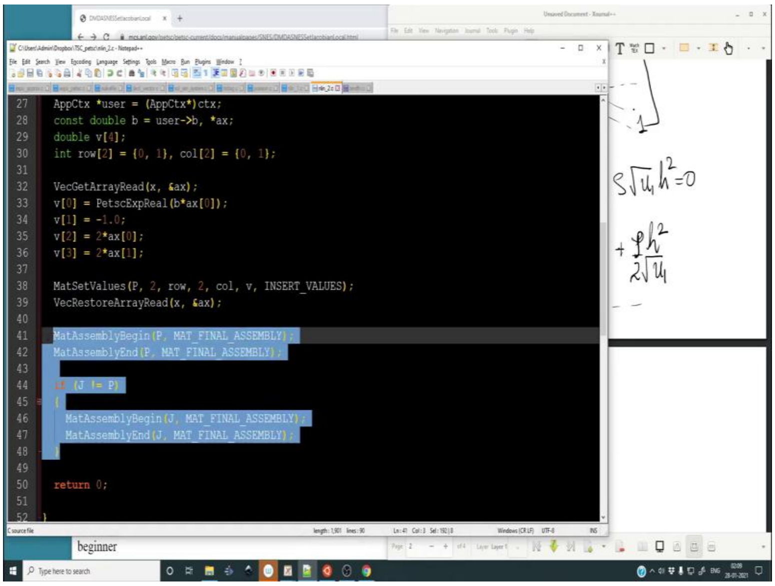


```
}

PetscErrorCode FormJac(DMDALocalInfo *info, double *au, Mat J, Mat P, AppCtx *user)
{
  int i;
  int col[3];
  double h = 1.0/(info->mx - 1);
  double v[3];

  for (i = info->xs; i < info->xs+info.xm; i++)
  {
    if ((i==0)||(i==info->mx-1))
    {
      v[0] = 1.0;
      MatSetValues(P, 1, &i, 1, &i, v, INSERT_VALUES);
    }
    else
    {
      col[0] = i-1; col[1] = i; col[2] = i+1;
      v[0] = -1.0; v[2] = -1.0; v[1] = 2+h*h*user->rho/(2*PetscSqrtReal(u[i]));
      MatSetValues(P, 1, &i, 3, col, v, INSERT_VALUES);
    }
  }


}
```

So, it is going to be 2+h*h*user->rho/(2*PetscSqrtReal(u[i])) alright. So, this is the Jacobian you do not need to insert these values obviously.

So, we set then we simply do MatSetValues into P we going to insert 1 row we going to pass the address of i, we going to insert 3 values we going to pass the column address we going to pass the values to be inserted address and insert values that is it. So, using this we have created the matrix P and we going to then have to assemble the matrix.

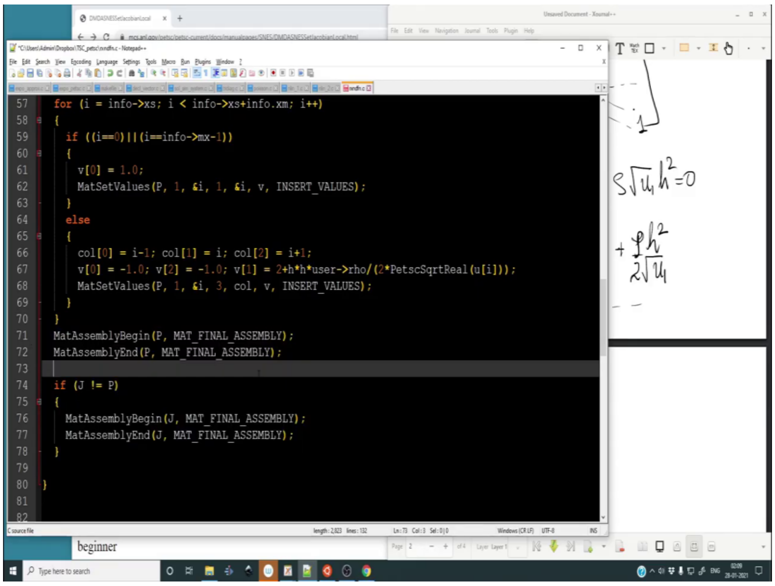
(Refer Slide Time: 41:03)



```
  AppCtx *user = (AppCtx*)ctx;
  const double b = user->b, *ax;
  double v[4];
  int row[2] = {0, 1}, col[2] = {0, 1};

  VecGetArrayRead(x, &ax);
  v[0] = PetscExpReal(b*ax[0]);
  v[1] = -1.0;
  v[2] = 2*ax[0];
  v[3] = 2*ax[1];

  MatSetValues(P, 2, row, 2, col, v, INSERT_VALUES);
  VecRestoreArrayRead(x, &ax);

  MatAssemblyBegin(P, MAT_FINAL_ASSEMBLY);
  MatAssemblyEnd(P, MAT_FINAL_ASSEMBLY);

  if (J != P)
  {
    MatAssemblyBegin(J, MAT_FINAL_ASSEMBLY);
    MatAssemblyEnd(J, MAT_FINAL_ASSEMBLY);
  }

  return 0;

}
```

So, we can borrow the program from the previous so in fact, we need this entire chunk.

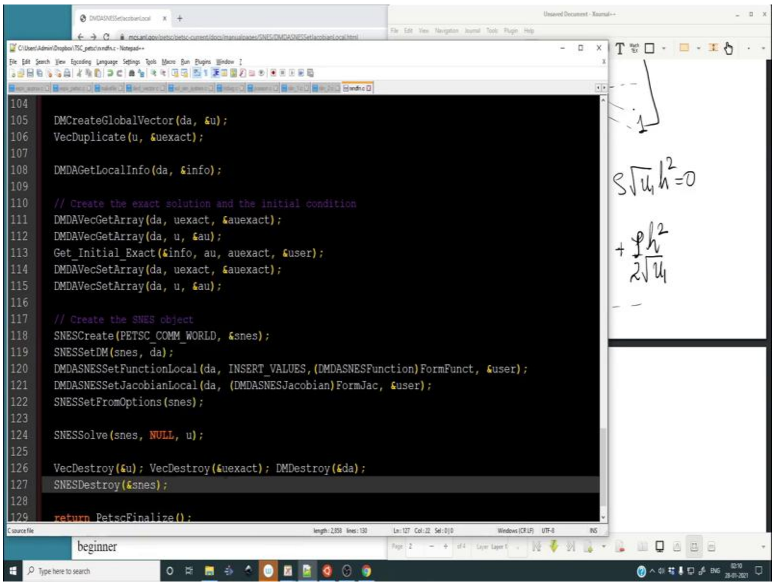
(Refer Slide Time: 41:09)



```
  for (i = info->xs; i < info->xs+info.xm; i++)
  {
    if ((i==0)||(i==info->mx-1))
    {
      v[0] = 1.0;
      MatSetValues(P, 1, &i, 1, &i, v, INSERT_VALUES);
    }
    else
    {
      col[0] = i-1; col[1] = i; col[2] = i+1;
      v[0] = -1.0; v[2] = -1.0; v[1] = 2+h*h*user->rho/(2*PetscSqrtReal(u[i]));
      MatSetValues(P, 1, &i, 3, col, v, INSERT_VALUES);
    }
  }
  MatAssemblyBegin(P, MAT_FINAL_ASSEMBLY);
  MatAssemblyEnd(P, MAT_FINAL_ASSEMBLY);

  if (J != P)
  {
    MatAssemblyBegin(J, MAT_FINAL_ASSEMBLY);
    MatAssemblyEnd(J, MAT_FINAL_ASSEMBLY);
  }

}
```

Because, either you assemble it if J != P then you assemble J whatever the user has provided ok alright. So, looks good. So, this is the way you call the functions this is syntax for it and after this we have made all this we simply then solve.

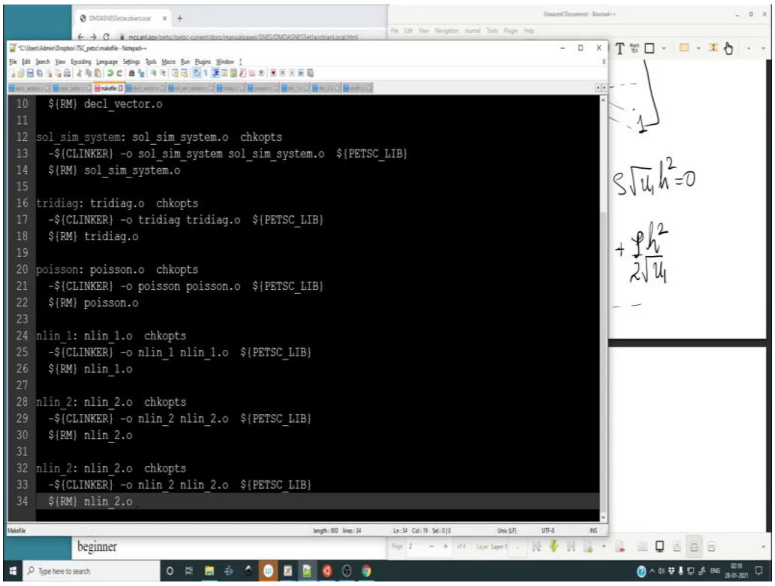
(Refer Slide Time: 41:34)



```

DMCreateGlobalVector(da, &u);
VecDuplicate(u, &uexact);

DMDAGetLocalInfo(da, &info);

// Create the exact solution and the initial condition
DMDAVecGetArray(da, uexact, &auexact);
DMDAVecGetArray(da, u, &au);
Get_Initial_Exact(&info, au, auexact, &user);
DMDAVecSetArray(da, uexact, &auexact);
DMDAVecSetArray(da, u, &au);

// Create the SNES object
SNESCreate(PETSC_COMM_WORLD, &snes);
SNESSetDM(snes, da);
DMDASNESSetFunctionLocal(da, INSERT_VALUES,(DMDASNESFunction)FormFunct, &user);
DMDASNESSetJacobianLocal(da, (DMDASNESJacobian)FormJac, &user);
SNESSetFromOptions(snes);

SNESSolve(snes, NULL, u);

VecDestroy(&u); VecDestroy(&uexact); DMDestroy(&da);
SNESDestroy(&snes);

return PetscFinalize();
```

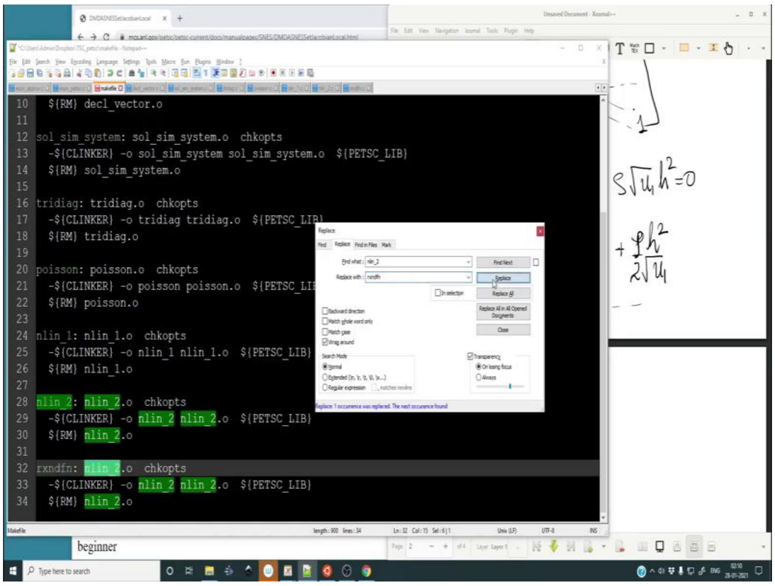
SNESSolve snes NULL put the solution in u and let us destroy whatever we had before finalizing so, VecDestroy(&uexact) then DMDestroy(&da) and SNESDestroy(&snes).

(Refer Slide Time: 42:19)



```
  ${RM} decl_vector.o

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

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

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

nlin_1: nlin_1.o  chkopts
  -${CLINKER} -o nlin_1 nlin_1.o  ${PETSC_LIB}
  ${RM} nlin_1.o

nlin_2: nlin_2.o  chkopts
  -${CLINKER} -o nlin_2 nlin_2.o  ${PETSC_LIB}
  ${RM} nlin_2.o

nlin_2: nlin_2.o  chkopts
  -${CLINKER} -o nlin_2 nlin_2.o  ${PETSC_LIB}
  ${RM} nlin_2.o
```

(Refer Slide Time: 42:28)



```
  ${RM} decl_vector.o

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

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

poisson: poisson.o  chkopts
  -${CLINKER} -o poisson poisson.o  ${PETSC_LI
  ${RM} poisson.o

nlin_1: nlin_1.o  chkopts
  -${CLINKER} -o nlin_1 nlin_1.o  ${PETSC_LIB}
  ${RM} nlin_1.o

nlin_2: nlin_2.o  chkopts
  -${CLINKER} -o nlin_2 nlin_2.o  ${PETSC_LIB}
  ${RM} nlin_2.o

rxndfn: nlin_2.o  chkopts
  -${CLINKER} -o nlin_2 nlin_2.o  ${PETSC_LIB}
  ${RM} nlin_2.o
```

Find what : nlin_2

Replace with : rxndfn

□ In selection

□ Backward direction

□ Match whole word only

□ Match case

☑ Wrap around

Search Mode

● Normal

○ Extended (\n, \r, \t, \0, \x...)

○ Regular expression □ . matches newline

☑ Transparency

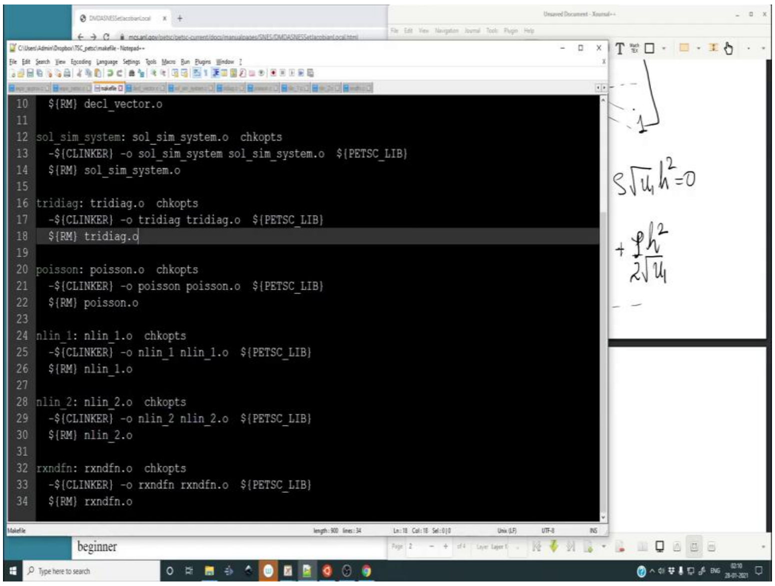
● On losing focus

○ Always

Replace: 1 occurrence was replaced. The next occurence found

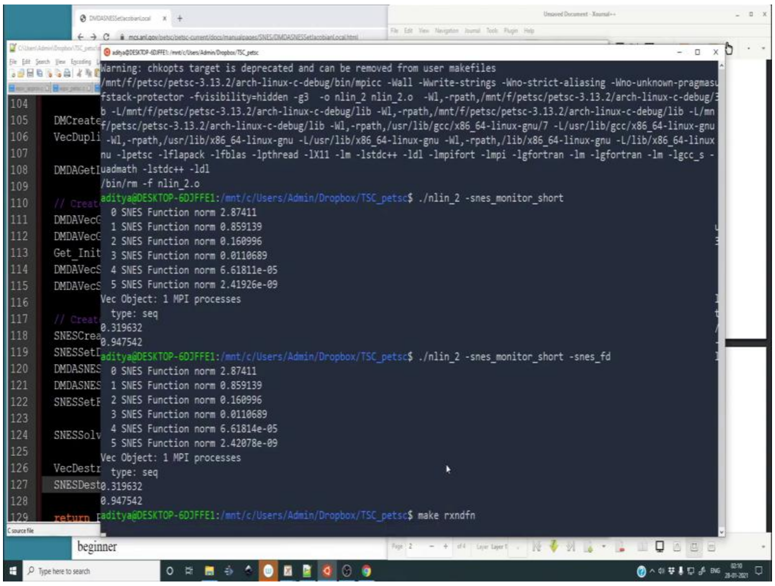
So, let us see whether this compiles or not. So let us create a new target so, rxn_dfn alright.

(Refer Slide Time: 42:39)



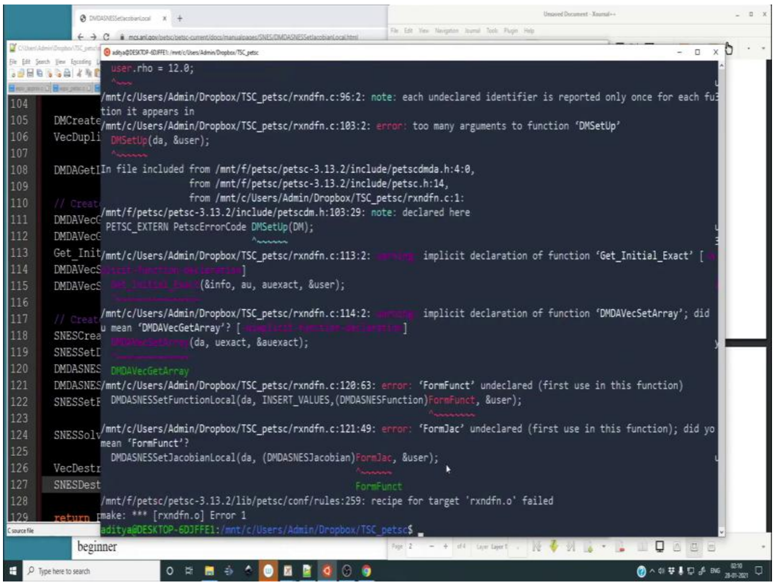
```
  ${RM} decl_vector.o

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

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

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

nlin_1: nlin_1.o  chkopts
  -${CLINKER} -o nlin_1 nlin_1.o  ${PETSC_LIB}
  ${RM} nlin_1.o

nlin_2: nlin_2.o  chkopts
  -${CLINKER} -o nlin_2 nlin_2.o  ${PETSC_LIB}
  ${RM} nlin_2.o

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

(Refer Slide Time: 42:43)



```
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-aliasing -Wno-unknown-pragmasu
fstack-protector -fvisibility=hidden -g3  -o nlin_2 nlin_2.o  -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/3
b -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mn
f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu
 -Wl,-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu -L/lib/x86_64-linux
nu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi -lgfortran -lm -lgfortran -lm -lgcc_s -
uadmath -lstdc++ -ldl
/bin/rm -f nlin_2.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./nlin_2 -snes_monitor_short
  0 SNES Function norm 2.87411
  1 SNES Function norm 0.859139
  2 SNES Function norm 0.160996
  3 SNES Function norm 0.0110689
  4 SNES Function norm 6.61811e-05
  5 SNES Function norm 2.41926e-09
Vec Object: 1 MPI processes
  type: seq
0.319632
0.947542
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./nlin_2 -snes_monitor_short -snes_fd
  0 SNES Function norm 2.87411
  1 SNES Function norm 0.859139
  2 SNES Function norm 0.160996
  3 SNES Function norm 0.0110689
  4 SNES Function norm 6.61814e-05
  5 SNES Function norm 2.42078e-09
Vec Object: 1 MPI processes
  type: seq
0.319632
0.947542
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make rxndfn
```

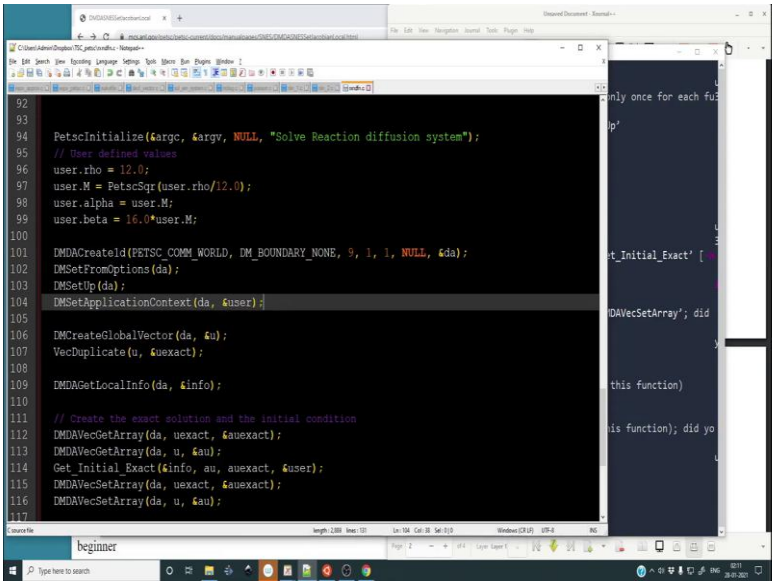
(Refer Slide Time: 42:47)



```
  user.rho = 12.0;
  ^~~~
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:96:2: note: each undeclared identifier is reported only once for each fu
tion it appears in
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:103:2: error: too many arguments to function ‘DMSetUp’
  DMSetUp(da, &user);
  ^~~~~~~
In file included from /mnt/f/petsc/petsc-3.13.2/include/petscdmda.h:4:0,
                 from /mnt/f/petsc/petsc-3.13.2/include/petsc.h:14,
                 from /mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:1:
/mnt/f/petsc/petsc-3.13.2/include/petscdm.h:103:29: note: declared here
 PETSC_EXTERN PetscErrorCode DMSetUp(DM);
                             ^~~~~~~
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:113:2: warning: implicit declaration of function ‘Get_Initial_Exact’ [-W
plicit-function-declaration]
  Get_Initial_Exact(&info, au, auexact, &user);
  ^~~~~~~~~~~~~~~~~
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:114:2: warning: implicit declaration of function ‘DMDAVecSetArray’; did
u mean ‘DMDAVecGetArray’? [-Wimplicit-function-declaration]
  DMDAVecSetArray(da, uexact, &auexact);
  ^~~~~~~~~~~~~~~
  DMDAVecGetArray
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:120:63: error: ‘FormFunct’ undeclared (first use in this function)
  DMDASNESSetFunctionLocal(da, INSERT_VALUES,(DMDASNESFunction)FormFunct, &user);
                                                               ^~~~~~~~~
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:121:49: error: ‘FormJac’ undeclared (first use in this function); did yo
mean ‘FormFunct’?
  DMDASNESSetJacobianLocal(da, (DMDASNESJacobian)FormJac, &user);
                                                 ^~~~~~~
                                                 FormFunct
/mnt/f/petsc/petsc-3.13.2/lib/petsc/conf/rules:259: recipe for target 'rxndfn.o' failed
make: *** [rxndfn.o] Error 1
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ _
```

So, I have saved this lets see whether it compiles ok there is an error form jac and all this ok let us see what the issues are so ok.

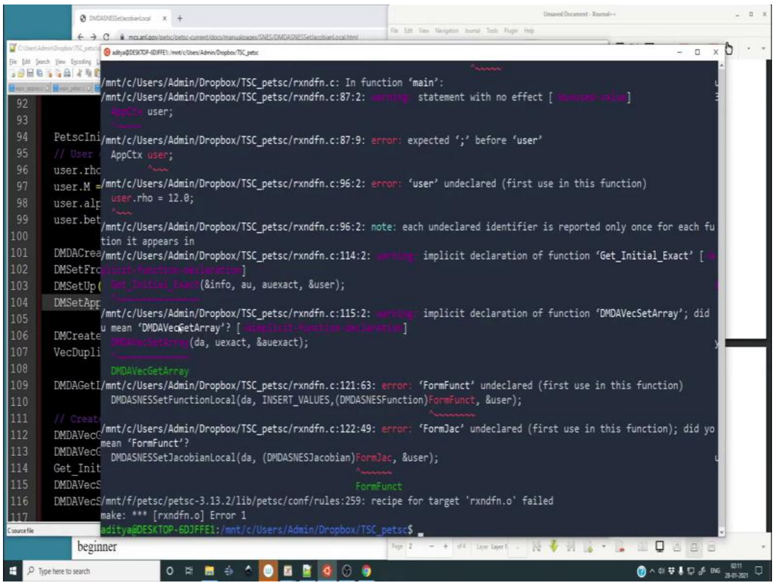
(Refer Slide Time: 43:14)



```
PetscInitialize(&argc, &argv, NULL, "Solve Reaction diffusion system");
// User defined values
user.rho = 12.0;
user.M = PetscSqr(user.rho/12.0);
user.alpha = user.M;
user.beta = 16.0*user.M;

DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &da);
DMSetFromOptions(da);
DMSetUp(da);
DMSetApplicationContext(da, &user);

DMCreateGlobalVector(da, &u);
VecDuplicate(u, &uexact);

DMDAGetLocalInfo(da, &info);

// Create the exact solution and the initial condition
DMDAVecGetArray(da, uexact, &auexact);
DMDAVecGetArray(da, u, &au);
Get_Initial_Exact(&info, au, uexact, &user);
DMDAVecSetArray(da, uexact, &auexact);
DMDAVecSetArray(da, u, &au);
```

So, DMSetUp does not require the user context we just need to set up the application context. So, DMSetUp DMSetApplicationContext dm and the address of user ok.

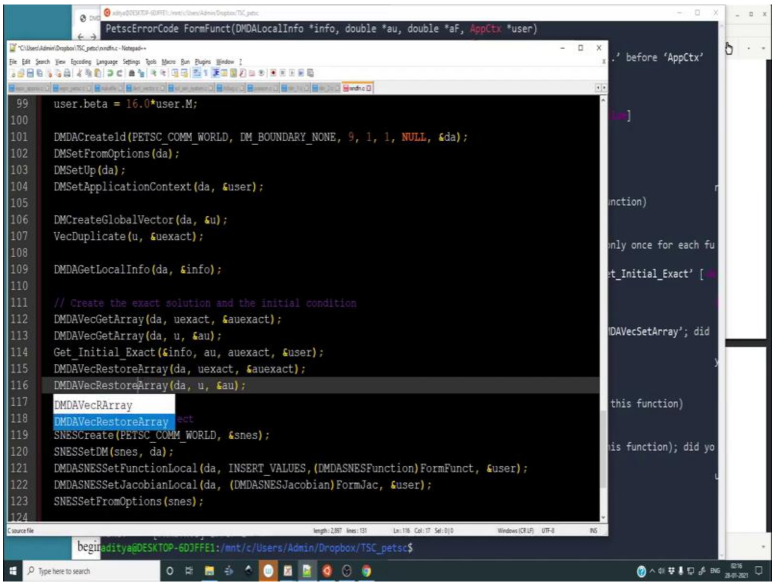
(Refer Slide Time: 43:36)



```
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c: In function ‘main’:
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:87:2: warning: statement with no effect [-Wunused-value]
  AppCtx user;
  ^~~~~~
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:87:9: error: expected ‘;’ before ‘user’
  AppCtx user;
         ^~~~
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:96:2: error: ‘user’ undeclared (first use in this function)
  user.rho = 12.0;
  ^~~~
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:96:2: note: each undeclared identifier is reported only once for each fu
tion it appears in
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:114:2: warning: implicit declaration of function ‘Get_Initial_Exact’ [-Wim
plicit-function-declaration]
  Get_Initial_Exact(&info, au, auexact, &user);
  ^~~~~~~~~~~~~~~~~
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:115:2: warning: implicit declaration of function ‘DMDAVecSetArray’; did
u mean ‘DMDAVecGetArray’? [-Wimplicit-function-declaration]
  DMDAVecSetArray(da, uexact, &auexact);
  ^~~~~~~~~~~~~~~
  DMDAVecGetArray
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:121:63: error: ‘FormFunct’ undeclared (first use in this function)
  DMDASNESSetFunctionLocal(da, INSERT_VALUES,(DMDASNESFunction)FormFunct, &user);
                                                               ^~~~~~~~~
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:122:49: error: ‘FormJac’ undeclared (first use in this function); did yo
mean ‘FormFunct’?
  DMDASNESSetJacobianLocal(da, (DMDASNESJacobian)FormJac, &user);
                                                 ^~~~~~~
                                                 FormFunct
/mnt/f/petsc/petsc-3.13.2/lib/petsc/conf/rules:259: recipe for target 'rxndfn.o' failed
make: *** [rxndfn.o] Error 1
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

So, let us see if that error goes away unexpected so, have we missed semicolon somewhere ok.

(Refer Slide Time: 44:00)



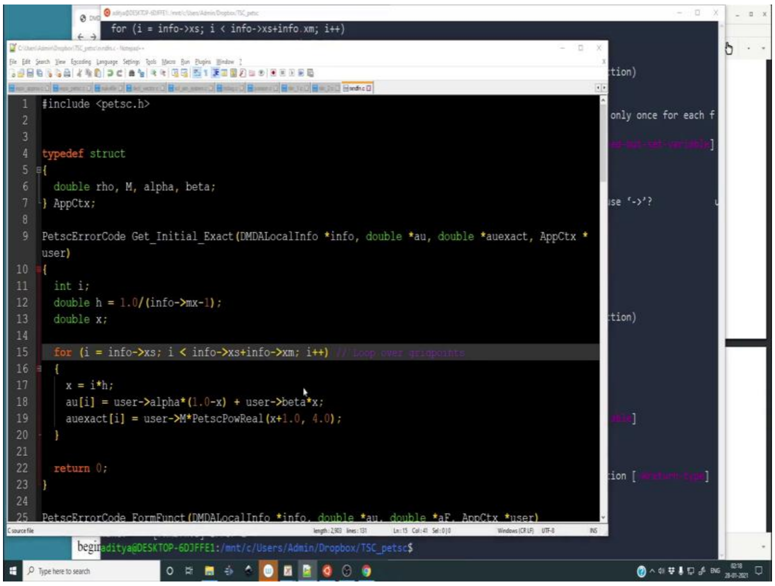
```
PetscErrorCode FormFunct(DMDALocalInfo *info, double *au, double *aF, AppCtx *user)
```

```
  user.beta = 16.0*user.M;

  DMDACreate1d(PETSC_COMM_WORLD, DM_BOUNDARY_NONE, 9, 1, 1, NULL, &da);
  DMSetFromOptions(da);
  DMSetUp(da);
  DMSetApplicationContext(da, &user);

  DMCreateGlobalVector(da, &u);
  VecDuplicate(u, &uexact);

  DMDAGetLocalInfo(da, &info);

  // Create the exact solution and the initial condition
  DMDAVecGetArray(da, uexact, &auexact);
  DMDAVecGetArray(da, u, &au);
  Get_Initial_Exact(&info, au, auexact, &user);
  DMDAVecRestoreArray(da, uexact, &auexact);
  DMDAVecRestoreArray(da, u, &au);

                         ect
  SNESCreate(PETSC_COMM_WORLD, &snes);
  SNESSetDM(snes, da);
  DMDASNESSetFunctionLocal(da, INSERT_VALUES,(DMDASNESFunction)FormFunct, &user);
  DMDASNESSetJacobianLocal(da, (DMDASNESJacobian)FormJac, &user);
  SNESSetFromOptions(snes);
```

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

There was a small mistake this is not SetArray this should be RestoreArray.

(Refer Slide Time: 44:18)



```
for (i = info->xs; i < info->xs+info.xm; i++)
```

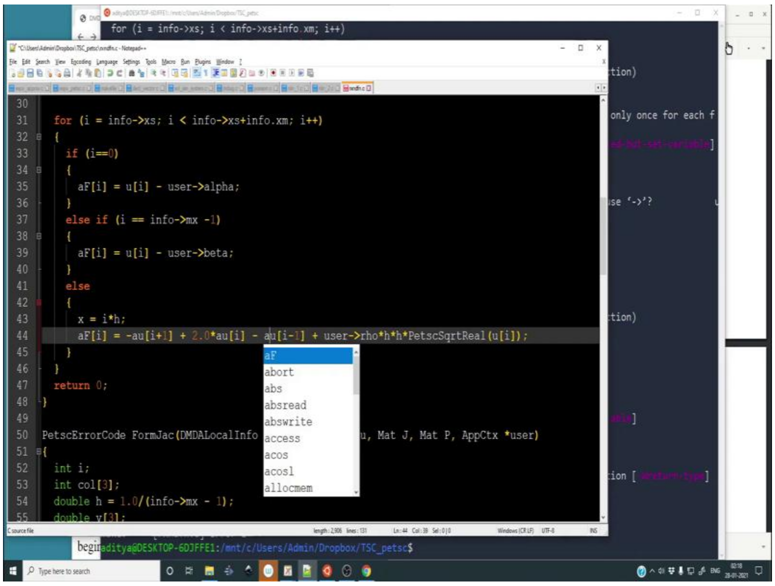
```
#include <petsc.h>


typedef struct
{
  double rho, M, alpha, beta;
} AppCtx;

PetscErrorCode Get_Initial_Exact(DMDALocalInfo *info, double *au, double *auexact, AppCtx *
user)
{
  int i;
  double h = 1.0/(info->mx-1);
  double x;

  for (i = info->xs; i < info->xs+info->xm; i++) // Loop over gridpoints
  {
    x = i*h;
    au[i] = user->alpha*(1.0-x) + user->beta*x;
    auexact[i] = user->M*PetscPowReal(x+1.0, 4.0);
  }

  return 0;
}

PetscErrorCode FormFunct(DMDALocalInfo *info, double *au, double *aF, AppCtx *user)
```

tion)
only once for each f
se '->'?
tion)
ion [
C source file
length : 2,903 lines : 131 Ln : 15 Col : 41 Sel : 0 | 0 Windows (CR LF) UTF-8 INS
Type here to search

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

Let me make it that is a small spelling mistake over here as well there is a small mistake over here as well.

(Refer Slide Time: 44:44)



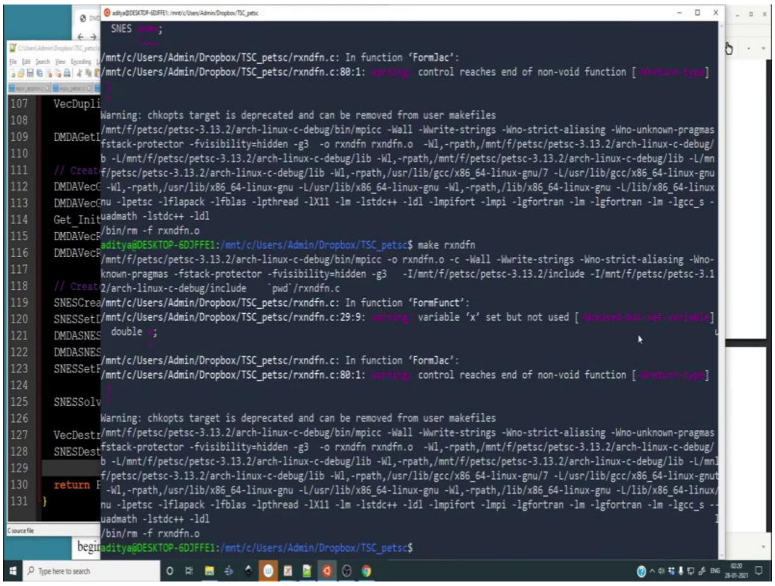
```
for (i = info->xs; i < info->xs+info.xm; i++)
```

```

  for (i = info->xs; i < info->xs+info.xm; i++)
  {
    if (i==0)
    {
      aF[i] = u[i] - user->alpha;
    }
    else if (i == info->mx -1)
    {
      aF[i] = u[i] - user->beta;
    }
    else
    {
      x = i*h;
      aF[i] = -au[i+1] + 2.0*au[i] - au[i-1] + user->rho*h*h*PetscSqrtReal(u[i]);
    }
  }
  return 0;
}

PetscErrorCode FormJac(DMDALocalInfo                u, Mat J, Mat P, AppCtx *user)
{
  int i;
  int col[3];
  double h = 1.0/(info->mx - 1);
  double v[3];
```

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

There is a small mistake over here as well. Well I have cleared some of the mistakes and they basically stemmed from declaring the variable inside the I mean as a function call to the declaring the variable going into the function as a u, but rather I am using u that was the error.

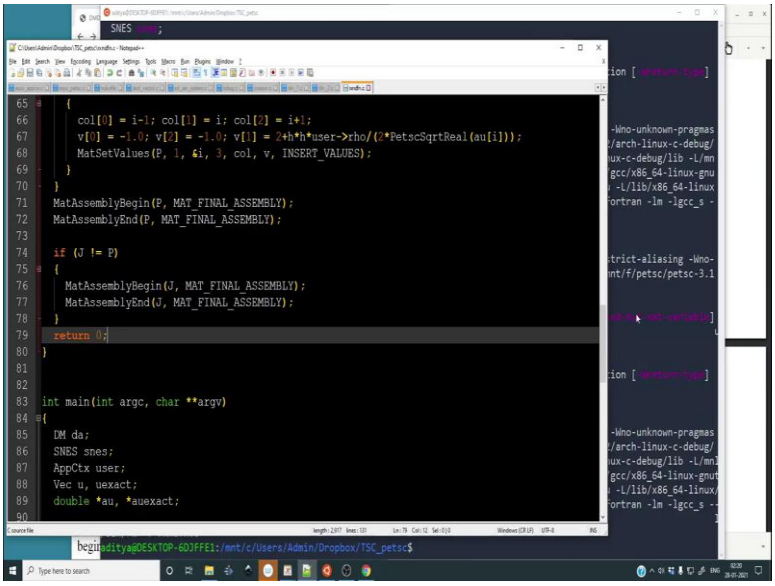
(Refer Slide Time: 45:10)



```
  SNES snes;

/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c: In function ‘FormJac’:
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:80:1: warning: control reaches end of non-void function [-Wreturn-type]
 }
 ^
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-aliasing -Wno-unknown-pragmas
fstack-protector -fvisibility=hidden -g3  -o rxndfn rxndfn.o  -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/
b -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mn
f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu
 -Wl,-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu -L/lib/x86_64-linux
nu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi -lgfortran -lm -lgfortran -lm -lgcc_s -
uadmath -lstdc++ -ldl
/bin/rm -f rxndfn.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make rxndfn
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o rxndfn.o -c -Wall -Wwrite-strings -Wno-strict-aliasing -Wno-
known-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/petsc/petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.1
2/arch-linux-c-debug/include    `pwd`/rxndfn.c
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c: In function ‘FormFunct’:
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:29:9: warning: variable ‘x’ set but not used [-Wunused-but-set-variable]
  double x;
         ^
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c: In function ‘FormJac’:
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:80:1: warning: control reaches end of non-void function [-Wreturn-type]
 }
 ^
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-aliasing -Wno-unknown-pragmas
fstack-protector -fvisibility=hidden -g3  -o rxndfn rxndfn.o  -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/
b -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mn
f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu
 -Wl,-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu -L/lib/x86_64-linux
nu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi -lgfortran -lm -lgfortran -lm -lgcc_s -
uadmath -lstdc++ -ldl
/bin/rm -f rxndfn.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

So, now I will remove the error so, in the functions control ok form jac we have not returned 0.

(Refer Slide Time: 45:27)

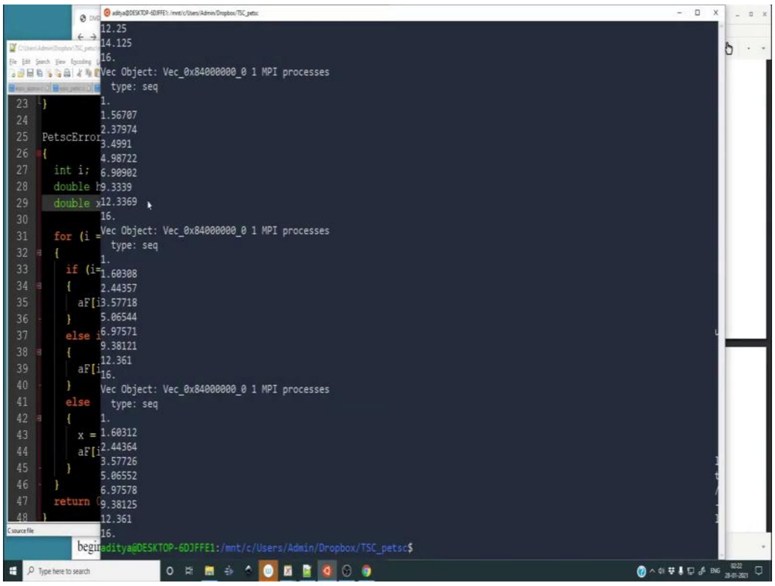


```
    {
      col[0] = i-1; col[1] = i; col[2] = i+1;
      v[0] = -1.0; v[2] = -1.0; v[1] = 2+h*h*user->rho/(2*PetscSqrtReal(au[i]));
      MatSetValues(P, 1, &i, 3, col, v, INSERT_VALUES);
    }
  }
  MatAssemblyBegin(P, MAT_FINAL_ASSEMBLY);
  MatAssemblyEnd(P, MAT_FINAL_ASSEMBLY);

  if (J != P)
  {
   MatAssemblyBegin(J, MAT_FINAL_ASSEMBLY);
   MatAssemblyEnd(J, MAT_FINAL_ASSEMBLY);
  }
  return 0;
}


int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;
  double *au, *auexact;
```

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

And what does it say x is set, but not used inside FormFunct ok.

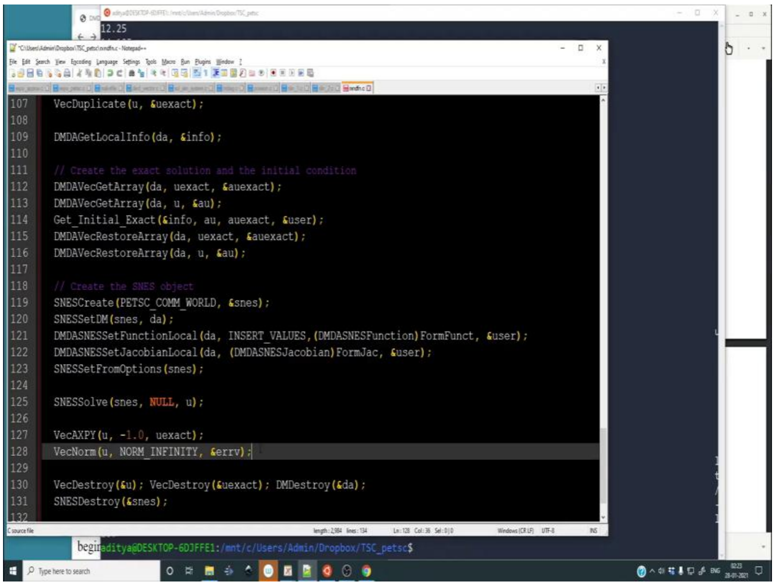
(Refer Slide Time: 45:44)



```
12.25
14.125
16.
Vec Object: Vec_0x84000000_0 1 MPI processes
  type: seq
1.
1.56707
2.37974
3.4991
4.98722
6.90902
9.3339
12.3369
16.
Vec Object: Vec_0x84000000_0 1 MPI processes
  type: seq
1.
1.60308
2.44357
3.57718
5.06544
6.97571
9.38121
12.361
16.
Vec Object: Vec_0x84000000_0 1 MPI processes
  type: seq
1.
1.60312
2.44364
3.57726
5.06552
6.97578
9.38125
12.361
16.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

So, we do seem to get convergence, but now we should now check it with the analytical solution. So, let us see how we can check it so, we at the end of this function we have the solution inside u.

(Refer Slide Time: 46:01)

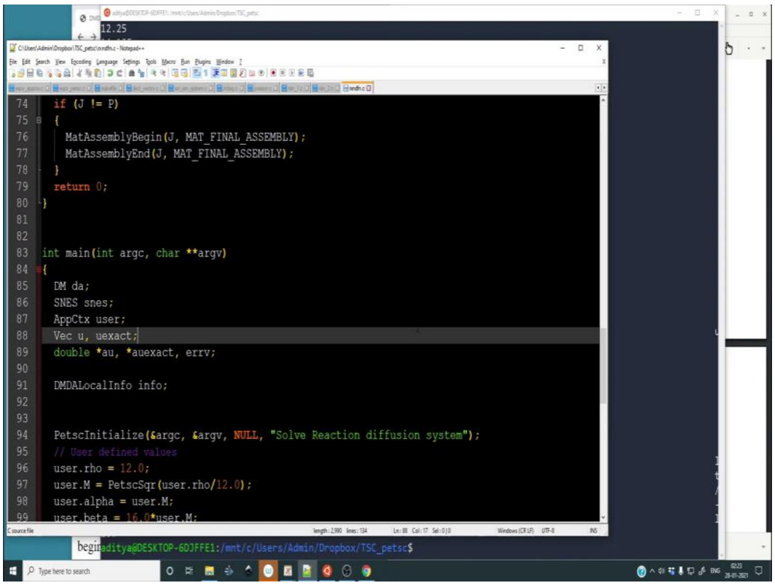


```
VecDuplicate(u, &uexact);

DMDAGetLocalInfo(da, &info);

// Create the exact solution and the initial condition
DMDAVecGetArray(da, uexact, &auexact);
DMDAVecGetArray(da, u, &au);
Get_Initial_Exact(&info, au, auexact, &user);
DMDAVecRestoreArray(da, uexact, &auexact);
DMDAVecRestoreArray(da, u, &au);

// Create the SNES object
SNESCreate(PETSC_COMM_WORLD, &snes);
SNESSetDM(snes, da);
DMDASNESSetFunctionLocal(da, INSERT_VALUES,(DMDASNESFunction)FormFunct, &user);
DMDASNESSetJacobianLocal(da, (DMDASNESJacobian)FormJac, &user);
SNESSetFromOptions(snes);

SNESSolve(snes, NULL, u);

VecAXPY(u, -1.0, uexact);
VecNorm(u, NORM_INFINITY, &errv);

VecDestroy(&u); VecDestroy(&uexact); DMDestroy(&da);
SNESDestroy(&snes);
```

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

So, what we must do is compare it with the exact solution. So, the exact solution was built over here. So, we will do the same thing `VecAXPY(u, -1.0, uexact)` and we will return the norm. So VecNorm(u, NORM_INFINITY, &errv) and the err variable has to be of kind double ok.

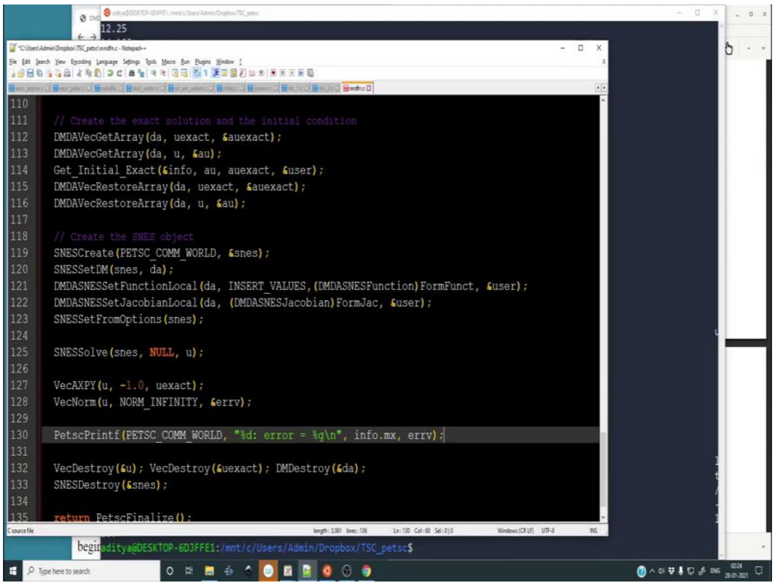
(Refer Slide Time: 46:47)



```
  if (J != P)
  {
    MatAssemblyBegin(J, MAT_FINAL_ASSEMBLY);
    MatAssemblyEnd(J, MAT_FINAL_ASSEMBLY);
  }
  return 0;
}


int main(int argc, char **argv)
{
  DM da;
  SNES snes;
  AppCtx user;
  Vec u, uexact;
  double *au, *auexact, errv;

  DMDALocalInfo info;


  PetscInitialize(&argc, &argv, NULL, "Solve Reaction diffusion system");
  // User defined values
  user.rho = 12.0;
  user.M = PetscSqr(user.rho/12.0);
  user.alpha = user.M;
  user.beta = 16.0*user.M;
```

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

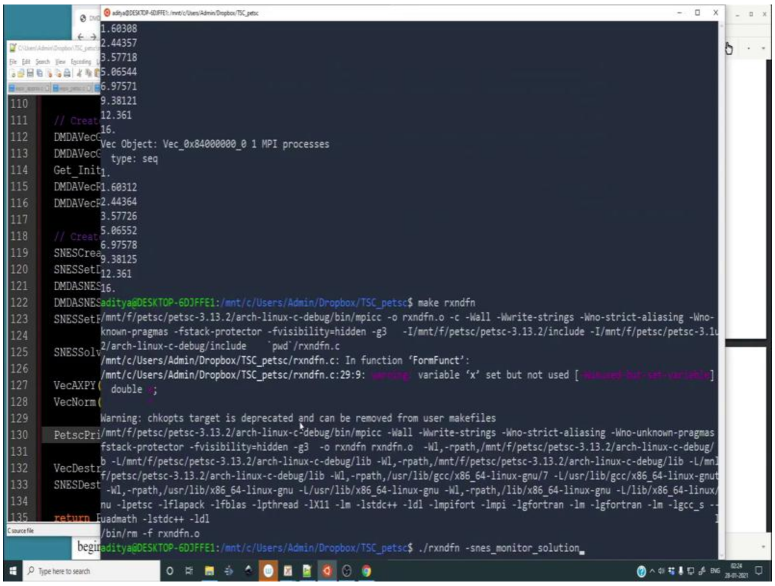
(Refer Slide Time: 46:59)



```
// Create the exact solution and the initial condition
DMDAVecGetArray(da, uexact, &auexact);
DMDAVecGetArray(da, u, &au);
Get_Initial_Exact(&info, au, auexact, &user);
DMDAVecRestoreArray(da, uexact, &auexact);
DMDAVecRestoreArray(da, u, &au);

// Create the SNES object
SNESCreate(PETSC_COMM_WORLD, &snes);
SNESSetDM(snes, da);
DMDASNESSetFunctionLocal(da, INSERT_VALUES,(DMDASNESFunction)FormFunct, &user);
DMDASNESSetJacobianLocal(da, (DMDASNESJacobian)FormJac, &user);
SNESSetFromOptions(snes);

SNESSolve(snes, NULL, u);

VecAXPY(u, -1.0, uexact);
VecNorm(u, NORM_INFINITY, &errv);

PetscPrintf(PETSC_COMM_WORLD, "%d: error = %g\n", info.mx, errv);

VecDestroy(&u); VecDestroy(&uexact); DMDestroy(&da);
SNESDestroy(&snes);

return PetscFinalize();
```

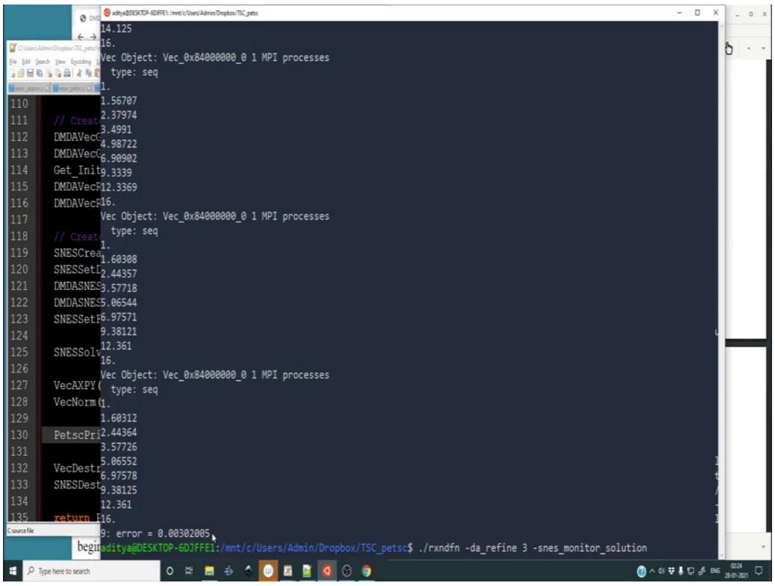
So, finally, we can print out the error. So, we can do a PetscPrintf it has to be done over the COMM WORLD and we will print %d, that is the number of grid points and the error as %g and this will be in info.mx, errv ok.

(Refer Slide Time: 47:35)



```
1.60308
2.44357
3.57718
5.06544
6.97571
9.38121
12.361
16.
Vec Object: Vec_0x84000000_0 1 MPI processes
  type: seq
1.
1.60312
2.44364
3.57726
5.06552
6.97578
9.38125
12.361
16.
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make rxndfn
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o rxndfn.o -c -Wall -Wwrite-strings -Wno-strict-aliasing -Wno-
known-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/petsc/petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.1
2/arch-linux-c-debug/include    `pwd`/rxndfn.c
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c: In function ‘FormFunct’:
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:29:9: warning: variable ‘x’ set but not used [-Wunused-but-set-variable]
  double x;

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-aliasing -Wno-unknown-pragmas
fstack-protector -fvisibility=hidden -g3  -o rxndfn rxndfn.o  -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/
b -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mn
f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu
 -Wl,-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu -L/lib/x86_64-linux/
nu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi -lgfortran -lm -lgfortran -lm -lgcc_s --
uadmath -lstdc++ -ldl
/bin/rm -f rxndfn.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -snes_monitor_solution
```

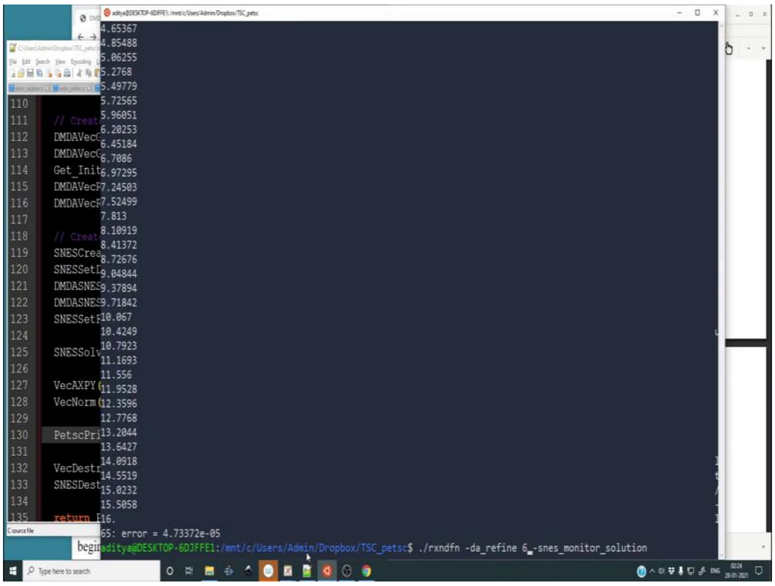
(Refer Slide Time: 47:38)



```
14.125
16.
Vec Object: Vec_0x84000000_0 1 MPI processes
  type: seq
1.
1.56707
2.37974
3.4991
4.98722
6.90902
9.3339
12.3369
16.
Vec Object: Vec_0x84000000_0 1 MPI processes
  type: seq
1.
1.60308
2.44357
3.57718
5.06544
6.97571
9.38121
12.361
16.
Vec Object: Vec_0x84000000_0 1 MPI processes
  type: seq
1.
1.60312
2.44364
3.57726
5.06552
6.97578
9.38125
12.361
16.
9: error = 0.00302005
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 3 -snes_monitor_solution
```

So, let us recompile ok so the error appears to be 0.003 let us do a grid refinement. So, because we are we have declared the default to be 9 we can do a minus da refine 3.

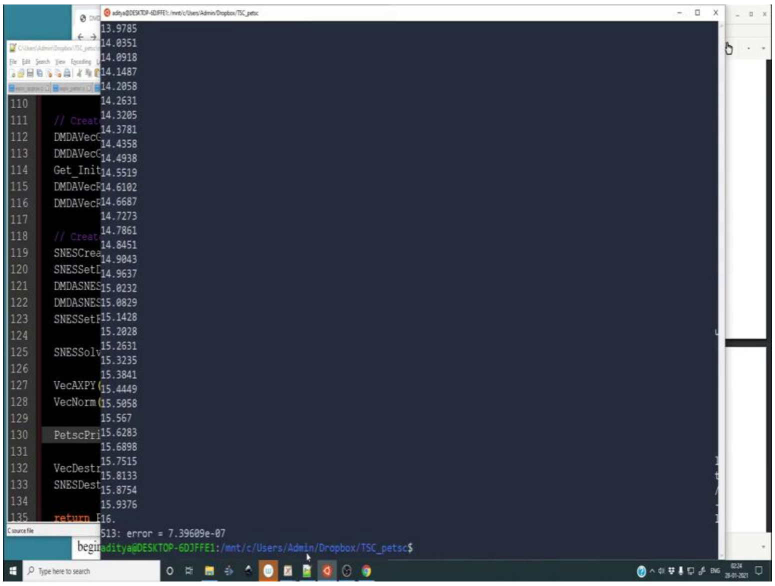
(Refer Slide Time: 47:59)



```
4.65367
4.85488
5.06255
5.2768
5.49779
5.72565
5.96051
6.20253
6.45184
6.7086
6.97295
7.24503
7.52499
7.813
8.10919
8.41372
8.72676
9.04844
9.37894
9.71842
10.067
10.4249
10.7923
11.1693
11.556
11.9528
12.3596
12.7768
13.2044
13.6427
14.0918
14.5519
15.0232
15.5058
16.
65: error = 4.73372e-05
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution
```

So, when we refine we get a much better norm let us do a larger refinement to see whether things are better ok.

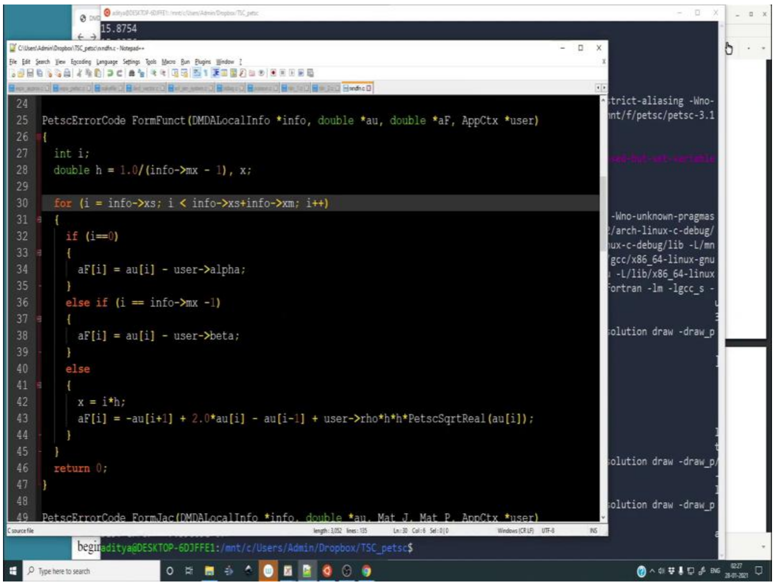
(Refer Slide Time: 48:15)



```
13.9785
14.0351
14.0918
14.1487
14.2058
14.2631
14.3205
14.3781
14.4358
14.4938
14.5519
14.6102
14.6687
14.7273
14.7861
14.8451
14.9043
14.9637
15.0232
15.0829
15.1428
15.2028
15.2631
15.3235
15.3841
15.4449
15.5058
15.567
15.6283
15.6898
15.7515
15.8133
15.8754
15.9376
16.
513: error = 7.39609e-07
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$
```

So, as we refine things become much better and we get conversions. So, this is the solution for some reason it says x is unused but, the fact of the matter is x is actually used I do not know why.

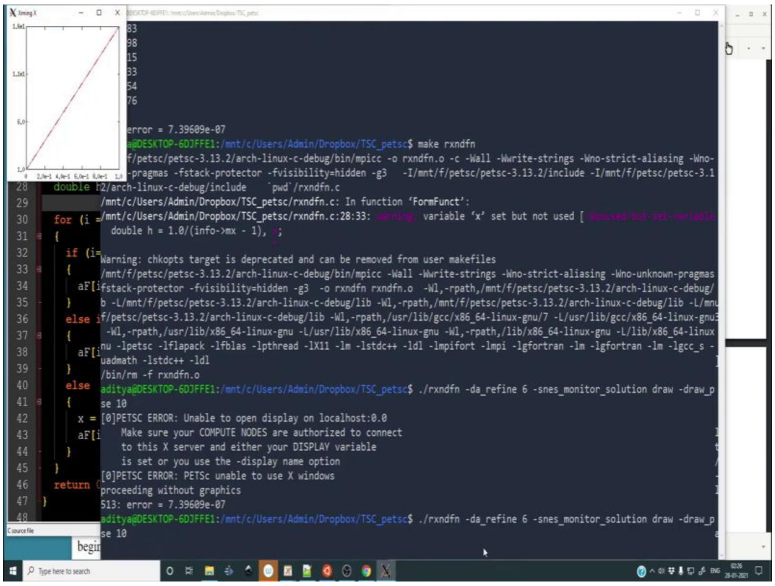
(Refer Slide Time: 48:36)



```
PetscErrorCode FormFunct(DMDALocalInfo *info, double *au, double *aF, AppCtx *user)
{
  int i;
  double h = 1.0/(info->mx - 1), x;

  for (i = info->xs; i < info->xs+info->xm; i++)
  {
    if (i==0)
    {
      aF[i] = au[i] - user->alpha;
    }
    else if (i == info->mx -1)
    {
      aF[i] = au[i] - user->beta;
    }
    else
    {
      x = i*h;
      aF[i] = -au[i+1] + 2.0*au[i] - au[i-1] + user->rho*h*h*PetscSqrtReal(au[i]);
    }
  }
  return 0;
}

PetscErrorCode FormJac(DMDALocalInfo *info, double *au, Mat J, Mat P, AppCtx *user)
```

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

(Refer Slide Time: 48:40)

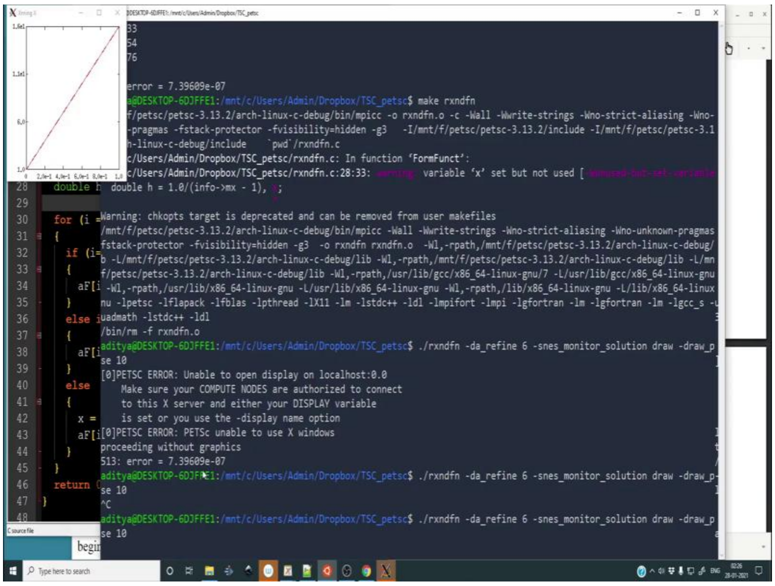


```
83
98
15
33
54
76

error = 7.39609e-07
a@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make rxndfn
f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o rxndfn.o -c -Wall -Wwrite-strings -Wno-strict-aliasing -Wno-
-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/petsc/petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.1
2/arch-linux-c-debug/include    `pwd`/rxndfn.c
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c: In function ‘FormFunct’:
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:28:33: warning: variable ‘x’ set but not used [-Wunused-but-set-variable
  double h = 1.0/(info->mx - 1), x;

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-aliasing -Wno-unknown-pragmas
fstack-protector -fvisibility=hidden -g3  -o rxndfn rxndfn.o  -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/
b -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mnt
f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu
 -Wl,-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu -L/lib/x86_64-linux
nu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi -lgfortran -lm -lgfortran -lm -lgcc_s -
uadmath -lstdc++ -ldl
/bin/rm -f rxndfn.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution draw -draw_p
se 10
[0]PETSC ERROR: Unable to open display on localhost:0.0
    Make sure your COMPUTE NODES are authorized to connect
    to this X server and either your DISPLAY variable
    is set or you use the -display name option
[0]PETSC ERROR: PETSc unable to use X windows
proceeding without graphics
513: error = 7.39609e-07
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution draw -draw_p
se 10
```

It is quite weird to see that error it is not an error it is a warning, but I do not like to see that warning anyway. So, the solution is written and well can we draw the; can we draw the solution.

Well let us see, whether we have the option to draw the solution let us run it with refinement 6 monitor solution draw pause 10 , we do not have x ok let us make let me start x min and now, we should be we should be able to show the ok.

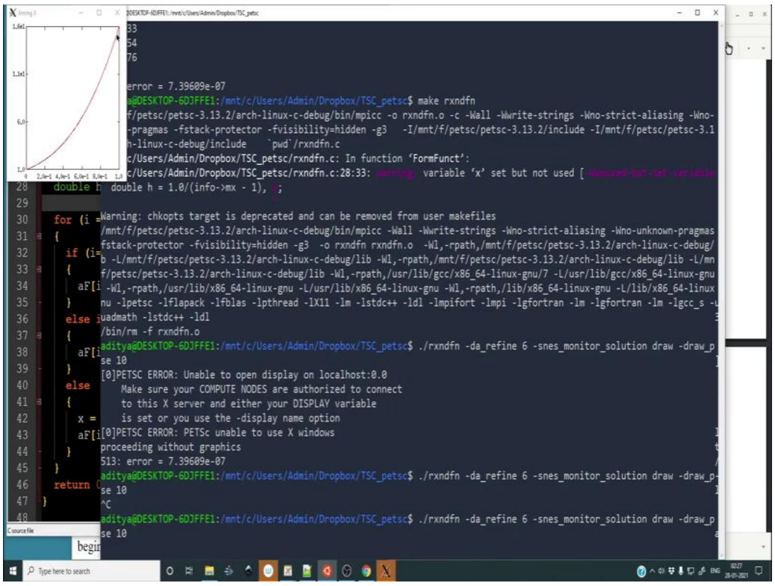
(Refer Slide Time: 49:50)



```
33
54
76

error = 7.39609e-07
a@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make rxndfn
f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o rxndfn.o -c -Wall -Wwrite-strings -Wno-strict-aliasing -Wno-
-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/petsc/petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.1
h-linux-c-debug/include    `pwd`/rxndfn.c
c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c: In function ‘FormFunct’:
c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:28:33: warning: variable ‘x’ set but not used [-Wunused-but-set-variable
  double h = 1.0/(info->mx - 1), x;

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-aliasing -Wno-unknown-pragmas
fstack-protector -fvisibility=hidden -g3  -o rxndfn rxndfn.o  -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/
b -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mn
f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu
 -Wl,-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu -L/lib/x86_64-linux
nu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi -lgfortran -lm -lgfortran -lm -lgcc_s -u
uadmath -lstdc++ -ldl
/bin/rm -f rxndfn.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution draw -draw_p
se 10
[0]PETSC ERROR: Unable to open display on localhost:0.0
    Make sure your COMPUTE NODES are authorized to connect
    to this X server and either your DISPLAY variable
    is set or you use the -display name option
[0]PETSC ERROR: PETSc unable to use X windows
proceeding without graphics
513: error = 7.39609e-07
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution draw -draw_p-
se 10
^C
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution draw -draw_p
se 10
```

So, the solution goes from it did go from 1 to 16 ok it goes from 1 to 16 well there you have it.

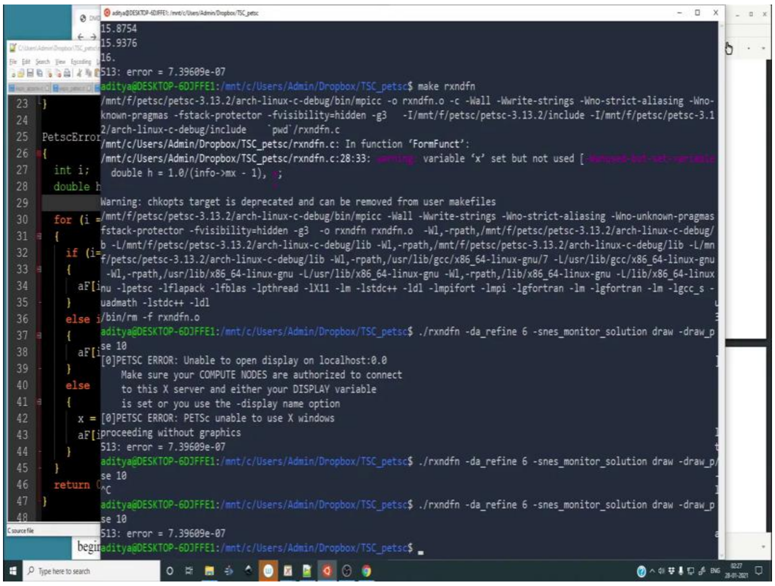
(Refer Slide Time: 50:03)



```
33
54
76

error = 7.39609e-07
a@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make rxndfn
f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o rxndfn.o -c -Wall -Wwrite-strings -Wno-strict-aliasing -Wno-
-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/petsc/petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.1
h-linux-c-debug/include    `pwd`/rxndfn.c
c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c: In function ‘FormFunct’:
c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:28:33: warning: variable ‘x’ set but not used [-Wunused-but-set-variable
 double h = 1.0/(info->mx - 1), x;

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-aliasing -Wno-unknown-pragmas
fstack-protector -fvisibility=hidden -g3  -o rxndfn rxndfn.o  -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/
b -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mn
f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu
 -Wl,-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu -L/lib/x86_64-linux
nu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi -lgfortran -lm -lgfortran -lm -lgcc_s -
uadmath -lstdc++ -ldl
/bin/rm -f rxndfn.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution draw -draw_p
se 10
[0]PETSC ERROR: Unable to open display on localhost:0.0
    Make sure your COMPUTE NODES are authorized to connect
    to this X server and either your DISPLAY variable
    is set or you use the -display name option
[0]PETSC ERROR: PETSc unable to use X windows
proceeding without graphics
513: error = 7.39609e-07
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution draw -draw_p
se 10
^C
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution draw -draw_p
se 10
```

Just like this we can solve a complicated problem it is updating the solution depending on the number of iterations and its going to take a while. So, for large number of refinements we do get a solution which is well converge with respect to the exact solution.

(Refer Slide Time: 50:33)



```
15.8754
15.9376
16.
513: error = 7.39609e-07
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ make rxndfn
/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/bin/mpicc -o rxndfn.o -c -Wall -Wwrite-strings -Wno-strict-aliasing -Wno-
known-pragmas -fstack-protector -fvisibility=hidden -g3   -I/mnt/f/petsc/petsc-3.13.2/include -I/mnt/f/petsc/petsc-3.1
2/arch-linux-c-debug/include    `pwd`/rxndfn.c
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c: In function ‘FormFunct’:
/mnt/c/Users/Admin/Dropbox/TSC_petsc/rxndfn.c:28:33: warning: variable ‘x’ set but not used [-Wunused-but-set-variable
 double h = 1.0/(info->mx - 1), x;

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-aliasing -Wno-unknown-pragmas
fstack-protector -fvisibility=hidden -g3  -o rxndfn rxndfn.o  -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/
b -L/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/mnt/f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -L/mn
f/petsc/petsc-3.13.2/arch-linux-c-debug/lib -Wl,-rpath,/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu
 -Wl,-rpath,/usr/lib/x86_64-linux-gnu -L/usr/lib/x86_64-linux-gnu -Wl,-rpath,/lib/x86_64-linux-gnu -L/lib/x86_64-linux
nu -lpetsc -lflapack -lfblas -lpthread -lX11 -lm -lstdc++ -ldl -lmpifort -lmpi -lgfortran -lm -lgfortran -lm -lgcc_s -
uadmath -lstdc++ -ldl
/bin/rm -f rxndfn.o
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution draw -draw_p
se 10
[0]PETSC ERROR: Unable to open display on localhost:0.0
    Make sure your COMPUTE NODES are authorized to connect
    to this X server and either your DISPLAY variable
    is set or you use the -display name option
[0]PETSC ERROR: PETSc unable to use X windows
proceeding without graphics
513: error = 7.39609e-07
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution draw -draw_p
se 10
^C
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ ./rxndfn -da_refine 6 -snes_monitor_solution draw -draw_p
se 10
513: error = 7.39609e-07
aditya@DESKTOP-6DJFFE1:/mnt/c/Users/Admin/Dropbox/TSC_petsc$ _
```

And in this particular lecture we have seen how to create out of a PDE in this case it was not a PDE but, in this case it was an ODE from that ODE we were able to create a sort of non-linear optimization problem, if you will we were able to construct the Jacobian of the discrete algebraic equations we were able to construct the functions of those equations and all those were done on a distributed grid.

So, do this on your own learn it by practice and you will see that all doing all this is not at all difficult; and with this I end this particular lecture I will see you again next time bye.