coursera COURSES ABOUT MARKELOV IGOR

UNIVERSITY of WASHINGTON
Scientific Computing

J. Nathan Kutz
University of Washington

- Home - Announcements
- Welcome to the Course
- Course Resources
- Video Lectures
- Quizzes
- Homework
- Discussion Forums
- Tips for Coursera Use
- FAQ about the Course
- Administration
- Join a Meetup
- Course Wiki

Forums / Assignments / **Homework 2-A**

Homework 2-A

homework2A × Add Tag Res

1 vote(s)

Students, what we should use solving this homework? Taylor series or Runge-Kutta 4th oder?

Markelov
on Mon 8 Oct 2012 6:31:1(

Add New Comment

Time (Oldest to Newest) Time (Newest to Oldest)

4 vote(s)

I am just getting started on HW 2-A and have some questions/comments. I would appreciate any feedback others can provide be
too far into this.

1. The assignment says to be sure to use forward and backward differencing for the boundaries, so I assume we are suppose
via the differencing equations and setting up A and b to solve Ax=b. Yes?
2. I don't see anything in the problem that states what order differencing equations to use, so I planned to use the seco
equations, (easier). Did I miss anything that suggested we should be using fourth order differencing equations?
3. As in HW 1, we don't have an explicit boundary condition, so I need to work on that. Plenty of notes from HW 1. Hopefully
something from that.
4. It looks like terms in the A matrix will contain the unknown eigenvalue, so I assume I need to make guesses at the eigen
iteratively solve Ax=b trying to get it to solve. Sound about right?
5. Finally, I don't recall anything from the lecture or notes which explained how to set up the A matrix in matlab. The general for
enough to understand, but if I understand correctly, we need to create something like an 80x80 matrix. I sure don't want to c
hand. I guess it's up to us to figure out how to create the matrix?

Thanks..

on Tue 9 Oct 201:

Add New Comment

2 vote(s)

Hi Anonymous,

1. the text of the Homework says "using a direct method" for this reason I think we should solve using the method exposed in p
1.5 of the lecture notes
2. Il think so
3. I think we should use the same boundary condition of HW1
4. I think so, but I'm not completely sure
5. We can use the spdiags function in Matlab. You need to provide a Matrix with the diagonals in the columns. Look at example
MATLAB documentation (spdiags documentation)

/
on Tue 9 Oct 201:

Add New Comment

2 vote(s)

Using the direct method I am obtaining very good approximations : 0.9999, 2.9999, 4.9999, 6.9999 , 8.9999; but these value
accepted by the grader. Then using again brute force I am obtaining three values accepted by the grader : 0.9994 (first eig
2.9968 (second eigenvalue) , 8.9506 (fifth eigenvalue). Please let me know if you are able to obtain the third eigenvalue and
eigenvalue. Many thanks. All the best.

on Wed 10 Oct 2012

Comments

2 What we should do? To write our own matrices A, x and b for 2 methods of HW A and B and just solve it x=A\B?

[Delete] Markelov Igor 

1 Hi Igor, try to implement your idea and please let me know if you obtain values accepted by the grader. Many thanks
and the best

lucas ochoa

1

Many thanks to Lucas for providing us with priceless information! :-)

Constantin Fishkin (Student)

Add New Comment

2
vote(s)

Hi Amedeo, Anonymous,

Regarding to your questions I think that:

1. Yes, it looks like we should use the method described in 1.5 "Boundary value problems: direct solve and relaxation" of th Notes
2. The hint in the quiz points on Table 4: (Second-order accurate forward- and backward-difference formulas) the Lecture N think we should use Table 4 for the boundary conditions and Table 2 (Second-order accurate center-difference formula discretizing of the equation itself.
3. The same hint in the quiz points on "grader approved" boundary condition in form y'(-L)=y(-L)*sqrt(L^2-e), y'(L)=-y(L)*sqrt(L^2
4. I think we should calculate matrix A(beta) and vector B, calculate y=A\B and select beta that gives best fit for y. What does mean is open question. It would be nice to discuss it.
5. Example of matrix A preparation is given in the Lecture Notes, p56 "Sparse Matrices: SPDIAG, SPY".

Constanti
on Fri 12 Oct 201:

Add New Comment

5
vote(s)

For the first version I used simple boundary condition on form of y(L)=y(-L)=a; Value of 'a' changes only scale and does normalized 'y'.

So the vector B is all zeros expect for B(1)=B(end)=a

Matrix A can be obtained from Table 2 (Second-order accurate center-difference formulas) and has 3 diagonals: the r $-[2+(x^2-\epsilon_n)*\Delta t]$, diagonals of 1s above it and below it.

As the criterion for ϵ_n I used the fact that *not* normalized solution y=A(ϵ)\B are much bigger when $\epsilon = \epsilon_n$ - kind or resonant. S follow the sum of squares - the bigger the sum is, the closer ϵ is to ϵ_n. And it it very sensitive - small change in ϵ causes huge the sum of squares.

As result I got: 0.999375305175781, 2.996893615722656, 4.992167358398437, 6.986931915283203, 8.989269104003904. Firs close to Lucas' values but the last - not. OK, probably it is due to simple boundary conditions

What I do not like here it is my criterion. It is sort of black magic. Does anybody have better idea for the criterion?

Constanti
on Fri 12 Oct 201:

Comments

3

Mr. Fishkin - I think you have a small typo above. The diagonal should be [-2 - (x^2 - en)dt] or equivalently -[2 + (x^2 - en)dt].

I was able to find values near 1, 5, 9 but not 3, 7. I had to find them brute force, ie, search every delta-beta interval from near zero through 10 and look for peaks. However, if you choose a large enough dbeta, there might not be a peak where you expect to find one. And searching with a small dbeta (0.0001) could take a few minutes. Do I have this about right, or have I gone down the wrong path?

Eric Pittelkau

3

Hi Constantin, your results are very close to the values that the grader is accepting. Now we have four values accepted by the grader: 0.9994 (first eigenvalue) , 2.9968 (second eigenvalue) , 4.9914 (third eigenvalue) , 8.9506 (fifth eigenvalue) . I will try to find the fourth eigenvalue. Many,many thanks Constantin, your are our leader. All the best.

lucas ochoa

2

Thank you, Lucas! Unfortunately the last value is far away from the grader approved so I afraid the fourth value will be hard to find :-(

Constantin Fishkin (Student)

3

Mr. Pittelkau, thank you for finding the typo, I have it corrected.

I started the iterations from the theoretical values 2*n-1 and used small delta. You should play with the bisection algorithm to make it working properly. But in the end it takes few seconds to find all the values and with good accuracy - I exit the cycle if delta step is less than 10^{-7}

But actually I asked for idea for different criterion. What do you think?

Constantin Fishkin (Student)

4

Hi Constantin, good news, thanks to your results now we have the five values accepted by the grader:

0.9994 (first eigenvalue) , 2.9968 (second eigenvalue) , 4.9914 (third eigenvalue) , 6.9804 (fourth eigenvalue) , 8.9506 (fifth eigenvalue)

I think that your results are correct, including the fifth eigenvalue and the grader has the error in the fifth eigenvalue.

Again, many, many thanks and all the best.

lucas ochoa

2

Viva Lucas!

Constantin Fishkin (Student)

4

I obtain matrix A with diagonal $-2 - x^2 dx$ (except [1,1] and [N, N] elements which i get from the boundary conditions from HW1 and using HINT) the part with ϵ I move to the other side of equation. As a result we have $Ax = \epsilon x$ - the standard eigenvalue problem. Then I use eig(A) and receive all the values which accepted by the grader. No iterations required.

Mikhail Garasyov (Student)

4

Good point. I thought about this way too. But the quiz asks for "using direct method" so I thought using standard procedure for eigenvalues will be cheating. :-) BTW did you use eig() or eigs() for sparse matrix?

Constantin Fishkin (Student)

5

I used eig. I think if I cheat then I do it in the same way as the grader does =)

Mikhail Garasyov (Student)

1

Dear Yue Hu, You are right but "asking what requirement you need for this matrix equation to have a non-trivial solution" is exactly looking for eigvalues - the way Mikhail did it. BTW "looking for resonant solution" in my approach is actually the same idea. Nevertheless Mikhail's approach is simpler and better.

Constantin Fishkin (Student)

1

Hi Constantin,

Yes, having non-trivial solution obviously is the same idea of eigenvalue. As Mikhail has stated, the [1,1] and [N,N] elements are not exactly multiplied by ϵ, so it is slightly different from exact eigenvalue problem. But I doubt that will make much of a difference.

If we don't try to use eigenvalue method, I think a simple method is the look at the determinant of the A matrix of the equation $Ax = 0$ and requires that the determinant to be zero (or very close to zero). This also yields the right answer very quickly.

There are ways to reframe the problem to make this an exact eigenvalue problem too.

Yue Hu (Student)

Add New Comment

This post has been deleted.

This post has been deleted.

1
vote(s)

This problem sure is giving me more trouble than HW-1. The technique sounds simple from the lectures, but as I work this probl running into questions I don't recall the lecture covering, so a couple more questions...

First I'm confused about when to use the forward and backward difference equations from Table 4. My intial approach was to equations for the first and last solution points. With that approach my A matrix is as described by Constantin except for the firs rows in which the diagonal is $2 - \Delta t^3 \left(x^2 - \varepsilon_n\right)$ with the next/previous several terms in the first/last row being -5, 4, and -1 bas difference equations for f'' in Table 4. When I review the lecture 5.4 from week 2 though, I notice that the forward and backward equations are not used at all. Rather the boundary conditions are used to modify the b vector. So I'm confused, the HW probler using the forward and backward differencing equations, but it seems the lectures did not use them. Can anyone explain when/h the Table 4 equations versus modifying b based on boundary conditions?

I eventually got somewhat reasonable values using the A matrix described above, (with forward and backward differencing equat for the first and last rows of A), but the error seemed to grow with each successive eigenvalue found. My approach was to eigenvalue by minimizing det(A) (I would increment/decrement the eigenvalue guess as long as abs(det(A)) was decreasing, ar the direction of adjustment when det(A) changed sign), and then solving y=A\b when det(A) < tol (my thinking was that I want trivial soluition to Ay=0, i.e. my b vector was the zero vector). As I said though, the error in eigenvalues grew with each s eigenvalue. Any thoughts on what the problem might be, or see any problems with the approach.

Thanks...

on Sun 14 Oct 201:

Add New Comment

1
vote(s)

I implemented the direct solution writing the matrix A with value $-[2 + \Delta x^2 * (kx^2 - \epsilon_n)]$ on principal diagonal and 1s on the a
below it. For testing the correct implementation of the matrix I used the eigenvalues provided by Constantin. The strange situat

when I set for example $\epsilon = 0.99938$ (or third or fifth eigenvalues) the result of y is as expected, instead when I set $\epsilon = 2.99689$
eigenvalue) the y is strange (no oscillation and with a very low norm). Do you have any ideas about this different behavior betwee
even eigenvalues?

on Sun 14 Oct 201:

Add New Comment

Reply to Thread *Use $$LaTeX$$ for math support.*

This forum supports the Markdown markup language. Markdown has some quirks with regard to line and paragraph breaks. By default, leaving one new line (pressing enter
anything. To start a new paragraph, leave two new lines. To force a line break with only one new line, leave two or more spaces after the end of the line.

- [ ] Make this post anonymous to other students.
- [x] Subscribe to this thread at the same time.

Reply to Thread

Instant Preview Render Math