6.3.2 Solving Lz = b (Forward substitution)

6.3.2 Solving Lz = b (Forward substitution)

places.

And we can recognize that U times x can be given a symbol y, the vector y.

And then if we first solve for y, then later we can solve for x.

And all I'm doing here is I happened to change my mind

and call the vector y vector z instead.

Let L be a unit lower triangular matrix.

What we want to do is solve Lz equals b, where z and b are vectors.

And this, of course, is the same as solving a lower triangular

system of equations because you can multiply L times z,

and from that then create a system of equations.

What we're going to do is

0:00 / 5:03 1.0x

we're going to use the exact

Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

Discussion

Hide Discussion

Topic: Week 6 / 6.3.2

Add a Post

Show all posts ▼ by recent activity ▼

There are no posts in this topic yet.

✖

Homework 6.3.2.1

1/1 point (graded)

Algorithm: $[b] := \text{LTRSV\_UNB\_VAR1}(L, b)$

Partition $L \rightarrow \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right)$, $b \rightarrow \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right)$
  where L_{TL} is 0×0, b_T has 0 rows
while $m(L_{TL}) < m(L)$ **do**
  Repartition

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} L_{00} & 0 & 0 \\ \hline l_{10}^T & \lambda_{11} & 0 \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \rightarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

$$b_2 := b_2 - \beta_1 l_{21}$$

  Continue with

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} L_{00} & 0 & 0 \\ \hline l_{10}^T & \lambda_{11} & 0 \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \leftarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

endwhile

Write the routine Ltrsv_unb_var1(L, b) that solves $Lx = b$, overwriting b. /p>

- `[ b_out ] = Ltrsv_unb_var1( L, b )`

You can check that they compute the right answers with the following script:

- test_Ltrsv_unb_var1.m (In `LAFF-2.0xM/Programming/Week06/`)

Unfortunately, PictureFLAME does not work for this problem.

This script exercises the functions by factoring the matrix

```
A = [
     2     0     1     2
    -2    -1     1    -1
     4    -1     5     4
    -4     1    -3    -8
]
```

by calling

```
LU = LU_unb_var5( A )
```

Next, it solves $Lz = b$ with the right-hand size vector

```
b = [
     2
     2
    11
    -3
]
```

by calling

```
z = Ltrsv_unb_var1( LU, b )
```

Finally, it extract upper triangular matrix U

```
U = triu( LU )
```

and solves $Ux = z$ with the intrinsic operation

```
x = U \ z
```

We can the check if this solves $Ax = b$ by computing

```
b - A * x
```

which should yield a zero vector.

- [x] Done/Skip ✔

✔

Here is our implementations of the function:

- Ltrsv_unb_var1.m

Submit

Answers are displayed within the problem