UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Lab 1

Implementing a solver for systems of linear equations

Julen Cayero, Cecilio Angulo

Bachelor's Degree in Video Game Design and Development

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Outline

1 Code hints

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Backtracking
Code hints

Simple pseudo-code for backtracking:

```
def backtracking(At,bt)

    1. Prealocate the vector x in memory. Size L x 1. L being the length of ...
         bt, cols or rows of At

    2. Solve for the last element of x

    3. Loop rows from the end-1 to the first row
        3.1 solve for x_r by:
        x_r = 1/diagonalTerm_r * (bt_r - At_r_{r+1:end} * x_{r+1 to end})

    4. Return x
```

Test:

$$\mathbf{A}_T = \begin{pmatrix} 5 & 4 & 2 \\ 0 & -3 & 1 \\ 0 & 0 & 45 \end{pmatrix} \quad \boldsymbol{b}_T = \begin{pmatrix} 5 \\ -5 \\ 45 \end{pmatrix} \quad \rightarrow \quad \boldsymbol{x} = \begin{pmatrix} -1 \\ 2 \\ 1 \end{pmatrix}$$

Naïve Forward Elimination

Code hints

Simple python code for triangulate a matrix:

```
def forward_elimination(A,b)

     1. Concatenation of A and B
     2. L = number of unknowns
     3. Loop p = #row from 0 to L-2
          4. Loop r = #rows from p+1 to L-1
               5. Identify pivot, pivot's row, subpivot and subpivot's row
               6. row_r = pivot*row_r - subpivot* row_p
     7. return At and Bt
```

Test:

$$\mathbf{A} = \begin{pmatrix} 5 & 4 & 2 \\ 2 & 1 & 1 \\ 1 & 2 & -3 \end{pmatrix} \quad \boldsymbol{b} = \begin{pmatrix} 5 \\ 1 \\ 0 \end{pmatrix} \quad \rightarrow \quad \mathbf{A}_T = \begin{pmatrix} 5 & 4 & 2 \\ 0 & -3 & 1 \\ 0 & 0 & 45 \end{pmatrix} \quad \boldsymbol{b}_T = \begin{pmatrix} 5 \\ -5 \\ 45 \end{pmatrix}$$