Discrete Boundary Value Problems

1 DISCRETE CALCULUS

I. Write the following matrices for boundary value problems with $h = 1/2$ on $[1,3]$ (use centered difference for $\frac{d}{dt}$):

(a) $\left[\frac{d^2}{dt^2}\right]\begin{cases}\text{fixed}\\\text{fixed}\end{cases}$ **(b)** $\left[\frac{d^2}{dt^2}\right]\begin{cases}\text{fixed}\\\text{free}\end{cases}$ **(c)** $\left[\frac{d^2}{dt^2}\right]\begin{cases}\text{free}\\\text{free}\end{cases}$ **(d)** $\left[\frac{d}{dt}\right]\begin{cases}\text{fixed}\\\text{fixed}\end{cases}$ **(e)** $\left[\frac{d}{dt}\right]\begin{cases}\text{free}\\\text{fixed}\end{cases}$

II. Write pointwise formulas for the following:

(a) y_n' forward difference **(b)** y_n' backward difference **(d)** y_n'' second difference **(c)** y_n' centered difference **(e)** y_n'''' *(for experts)*

2 FROM BVPS TO MATRIX EQUATIONS (BASIC)

Convert the following Boundary Value Problems (BVPs) into matrix equations:

(**a**) $y'' = 1, \quad y(0) = y(4) = 0, \quad h = 1.$

(**b**) $y'' = t, \quad y(0) = y(1) = 0, \quad h = 1/4.$

(**c**) $y'' + y = \cos(t), \quad y(0) = y(\pi) = 0, \quad h = \pi/4.$

(**d**) $y'' + ty = \delta(t), \quad y(-1) = y(1) = 0, \quad h = 1/2.$

3 FROM BVPS TO MATRIX EQUATIONS (MEDIUM)

Convert the following Boundary Value Problems (BVPs) into matrix equations:

(**a**) $y'' = t^2, \quad y(0) = 2, \; y(4) = 0, \quad h = 1$

(**b**) $y'' = 2t, \quad y(0) = 5, \; y(1) = -6, \; h = 1/5$

(**c**) $y'' = t, \quad y'(-1) = 0, \; y(3) = 0, \quad h = 1$

(**d**) $y'' = \sin(2t), \quad y'(0) = 0, \; y'(2\pi) = 0, \; h = \pi/3$

(**e**) $y'' + 4ty = \delta(t), \quad y'(-1) = 0, \quad y'(1) = 0, \; h = 1/2$

(**f**) $y'' + \delta(t - \pi/2)y = 1, \quad y(0) = 5, \quad y(\pi) = 5, \; h = \pi/4$

4 FROM BVPS TO MATRIX EQUATIONS (ADVANCED)

Convert the following Boundary Value Problems (BVPs) into matrix equations for $h = 1/2$.
Think about how to define the resulting matrices when h becomes small and the matrix sizes grow large.

(a) $y'' = 1, \quad y'(-3) = 5, \; y'(-1) = -8.$

(b) $y'' + y' = 1, \quad y(-2) = 0, \quad y(0) = 0.$

(c) $y'' + 4ty' + 2y = 1, \quad y(-2) = 0, \quad y(0) = 0.$

(d) $y'' + 4ty' = 2t, \quad y(0) = 2, \; y(2) = -3.$

(e) $y'' + 2y' + ty = 3, \quad y(1) = 5, \; y(3) = 7.$

(f) $y'' + 2y' + ty = 3, \quad y'(1) = 5, \; y'(3) = 7.$

5 DISCRETE IMPULSE RESPONSE AND SOLUTIONS TO DE

A differential equation $\mathbf{D}[y] = f(t)$ with $\begin{cases} y(-1) = 0 \\ y'(1) = 0 \end{cases}$ has the following impulse responses with $h = \frac{2}{5}$.

$$\boldsymbol{y}^{(1)} = \begin{bmatrix}1\\0\\2\\1\end{bmatrix} \qquad \boldsymbol{y}^{(2)} = \begin{bmatrix}0\\1\\0\\-1\end{bmatrix} \qquad \boldsymbol{y}^{(3)} = \begin{bmatrix}1\\0\\1\\1\end{bmatrix} \qquad \boldsymbol{y}^{(4)} = \begin{bmatrix}2\\1\\-1\\0\end{bmatrix}$$

Write the solution to $\mathbf{D}[y] = 3t + 1$ with the same boundary values and h.

Getting a good approximation from discretizing differential equations requires very small step-sizes, which results in very large matrices – discretizing with $h = 0.01$ results in a 99×99 matrix for just an interval of length 1! This week we will learn about how to make the kinds of big matrices needed to discretize differential equations with small h.

- The five basic MatLab commands for creating big matrices are:

```
zeros(<#rows>,<#cols>)        eye(<size>)            diag(<vector>,<diag #>)
ones(<#rows>,<#cols>)                                toeplitz(<col 1>,<row 1>)
```

  The commands `zeros` and `ones` create matrices of all 0 or 1. These are most useful for making 0 or 1 vectors, for example `zeros(1,100)` makes a row vector of one hundred 0's and `ones(100,1)` makes a column vector of one hundred 1's.
  The commands `eye` and `diag` make identity (eyedentity) and diagonal matrices.
  For us, the most useful command is `toeplitz`, which makes matrices with constant diagonals copying specified column and row values. If only one vector is supplied, it is used for both the first row and column.

```
>> toeplitz( [1  5  7  9], ... % col 1
             [1  0 -2 -4]  )   % row 1

ans =
     1     0    -2    -4
     5     1     0    -2
     7     5     1     0
     9     7     5     1
```

```
>> toeplitz( [-2 1 zeros(1,2)] )

ans =
    -2     1     0     0
     1    -2     1     0
     0     1    -2     1
     0     0     1    -2
```

- We can use `toeplitz` to discretize $y'' + y' + y = t^2 + 1$, $y(0) = 0$, $y(10) = 0$ with $h = 0.01$ and then compute the discrete solution.

```
>> h = 0.01;
>> t = 0+h : h : 10-h;                      % t_1 ... t_n              (middle t-values)
>> n = length(t);                           % use n x n matrices to get  y_1 ... y_n
>> D2 = toeplitz( [-2 1 zeros(1,n-2)] );    % 2nd derivative matrix  y"  (fixed-fixed)
>> D1 = toeplitz( [0 -1 zeros(1,n-2)], ...  % 1st derivative matrix  y'  (fixed-fixed)
                  [0  1 zeros(1,n-2)] );    %   (using centered differences)
>> D0 = eye(n);                             % 0th derivative matrix  y
>> f  = t.^2 + 1;                           % discrete forcing function: f = t^2 + 1
>> DE = 1/h^2 * D2  +  1/(2*h) * D1  +  D0; % differential equation matrix
>> y = DE \ f' ;                            % discrete solution to differential eqn
>> plot (t, y)                              % graph the solution to differential eqn
```

- For a more difficult example, we can solve $y'' + 2t\,y' + t^2\,y = t^3$, $y'(0) = 0$, $y(10) = 0$ with $h = 0.01$.

```
>> h = 0.01;
>> t = 0+h : h : 10-h;                      % t_1 ... t_n              (middle t-values)
>> n = length(t);                           % use n x n matrices to get  y_1 ... y_n
>> D2 = toeplitz( [-2 1 zeros(1,n-2)] );    % 2nd derivative matrix  y"  (fixed-fixed)
>> D2(1,1) = -1;                            %            |----> convert to (free-fixed)
>> D1 = toeplitz( [0 -1 zeros(1,n-2)], ...  % 1st derivative matrix  y'  (fixed-fixed)
                  [0  1 zeros(1,n-2)] );    %   (using centered differences)
>> D1(1,1) = -1;                            %            |----> convert to (free-fixed)
>> D0 = eye(n);                             % 0th derivative matrix  y
>> P = diag(2*t);                           % "multiply by 2t"   matrix
>> Q = diag(t.^2);                          % "multiply by t^2"  matrix
>> f  = t.^3;                               % discrete forcing function:  f = t^3
>> DE =         1/h^2 * D2    +   ...       % eqn =  y" +
          P * (1/(2*h) * D1)  +   ...       %        p y' +
          Q       *      D0;                %        q y
>> y = DE \ f' ;                            % discrete solution to differential eqn
>> plot (t, y)                              % graph the solution to differential eqn
```