Name: Ibrahim YAZICI

ID: 4367941

FINAL PROJECT

1) The Shooting Method

* The file shooting_method.m contains the code that solves the problem by the shooting method.
* The file func_shooting_method.m contains the code that solves the ivp in the shooting method by AB2 method.
* Here are the step sizes and corresponding maximum errors that are found by absolute value of difference between true solution (uexct in code) and numerical solution (uappl in the code)

h(hstep)	max_error
$\frac{1}{20}$ $\longrightarrow$	0.1153
$\frac{1}{40}$ $\longrightarrow$	0.1416
$\frac{1}{80}$ $\longrightarrow$	0.0926
$\frac{1}{160}$ $\longrightarrow$	0.0363
$\frac{1}{320}$ $\longrightarrow$	0.0119

We observe that, when we divide the step-size by 2, maximum error is divided by ≈ 4, after $h=\frac{1}{80}$. This result is consistent with the fact that AB2 is of order 2.

Also, we know that the ivp we solve in the shooting method is unstable, that might be the reason that max_error is increased between $h=\frac{1}{20}$ and $h=\frac{1}{40}$.

2) The Symmetric Finite Difference Method

* The file symmetric_fdm.m contains the code that solves the problem by the symmetric finite difference method.
* Here are the step sizes and corresponding maximum errors that are found by absolute value of difference between the true solution (uexact in the code) and the numerical solution (uapp in the code)

h (hstep in code)		Max_error
$\frac{1}{20}$	$\longrightarrow$	0.4353
$\frac{1}{40}$	$\longrightarrow$	0.1932
$\frac{1}{80}$	$\longrightarrow$	0.0557
$\frac{1}{160}$	$\longrightarrow$	0.0121

We observe that, from $h=\frac{1}{80}$ to $h=\frac{1}{160}$, when the h is divided by 2, the maximum error is divided by ≈ 4, which is consistent with the fact that the symmetric FDM is of order 2.

* Also, we observe that for $h=\frac{1}{20}$ and $h=\frac{1}{40}$, since the problem is stiff, there are oscillations in the stiff region, as expected in the implementation of the symmetric FDM for stiff problems

3) The Upwind Finite Difference Method

* The file upwind_fdm.m contains the code that solves the problem by the upwind finite difference method.

* Here are the step sizes and maximum errors as described in the previous applications

h (hstep in the code)		Max_error
$\frac{1}{20}$	$\longrightarrow$	0.1599
$\frac{1}{40}$	$\longrightarrow$	0.2036
$\frac{1}{80}$	$\longrightarrow$	0.1579
$\frac{1}{160}$	$\longrightarrow$	0.0922
$\frac{1}{320}$	$\longrightarrow$	0.0517

We observe that, when h is divided by 2, the maximum error is divided by ≈ 2, which is consistent with the fact that the upwind FDM is of order 1.

⊛ Also, unlike the symmetric FDM, we observe that the numerical solution does not display oscillations for any value of h in the stiff region, which is consistent with the fact that upwind FDM is absolutely stable for any h.

4) Error Estimation and Adaptive Mesh Refinement

⊛ The file nonuniform_adaptive_fdm.m contains the code that solves the problem by FDM employed in error estimation and adaptive mesh refinement.

⊛ The file func_nonuniform_adaptive_fdm.m contains the code that solves the problem by FDM by nonuniform mesh. This function is used in the process of error estimation and adaptive mesh refinement.

⊛ When we run the code for all the values of $h = \frac{1}{20}, \frac{1}{40}, \frac{1}{80}, \frac{1}{160}$, we observe that max_error = 0.0030 in each case, which is expected because during the process of error estimation and adaptive mesh refinement, the the values in non-uniform mesh are created to be more dense in stiff region, which results a very good approximation compared to the previous methods. Also, we should note that, the final nonuniform mesh for all values of h are same.

⊛ Also, we can express that when max_error's of all the methods are compared, we can observe that error estimation and adaptive mesh refinement gives a very strong tool to get a nice numerical solution.

⊛ Finally, we should emphasize that max_error values above are obtained when ETOL = 0.01 However, if we take ETOL smaller (ETOL = 0.001 for example), we see that max_error values are much more smaller and we obtain a much better numerical approximation.