3. Multigrid Method

In this problem, we implement a multigrid method for the differential equation

$$u'' = f, \qquad u(0) = 0, \qquad u(1) = 0$$

with u and f given by (4.82) and (4.83) in the book. Note that this time we use zero boundary values.

Some extra details about the method can be found e.g. in Gilbert Strang: Mathematical Methods for Engineers II, pages one and two.

The main part of the method is a function (or alternatively/better a member of a class)

```
step(U0, RHS, level):
```

which performs one V-cycle of the multigrid method for the system `A U = RHS` with initial value `U0`. The V-cycle is the algorithm we had in class and is summarized on p. 107 of LeVeque's book.

1. For steps 1. and 6. (in the book), we use $\nu = 3$ iterations of an underrelaxed Jacobi method with $\omega = 2/3$.

2. For the coarsening/restriction and interpolation in steps 2. and 5., you can use the matrices `R` and `I` in Strang's lecture notes. Finally, the restricted matrix $\tilde{A}$ in step 4. is given by $\tilde{A} = RAI$.

3. In step 4. the function `step` calls itself with `level-1` as new level. This is called recursion in programming. The level corresponds to the number of grid points via $m = 2^{level} - 1$. Note that we have 2^{level} intervals and $2^{level} - 1$ interior grid points. For `level=1` we terminate this process and use a python solver to solve the remaining linear system, e.g. `scipy.linalg.spsolve`.

Make a loglog plot of the number of V-cycles versus the error (you can get the exact solution of the linear system via `scipy.linalg.spsolve`) for $m = 63 = 2^6 - 1$ gird points and 300 iterations of both the multigrid and Jacobi methods.