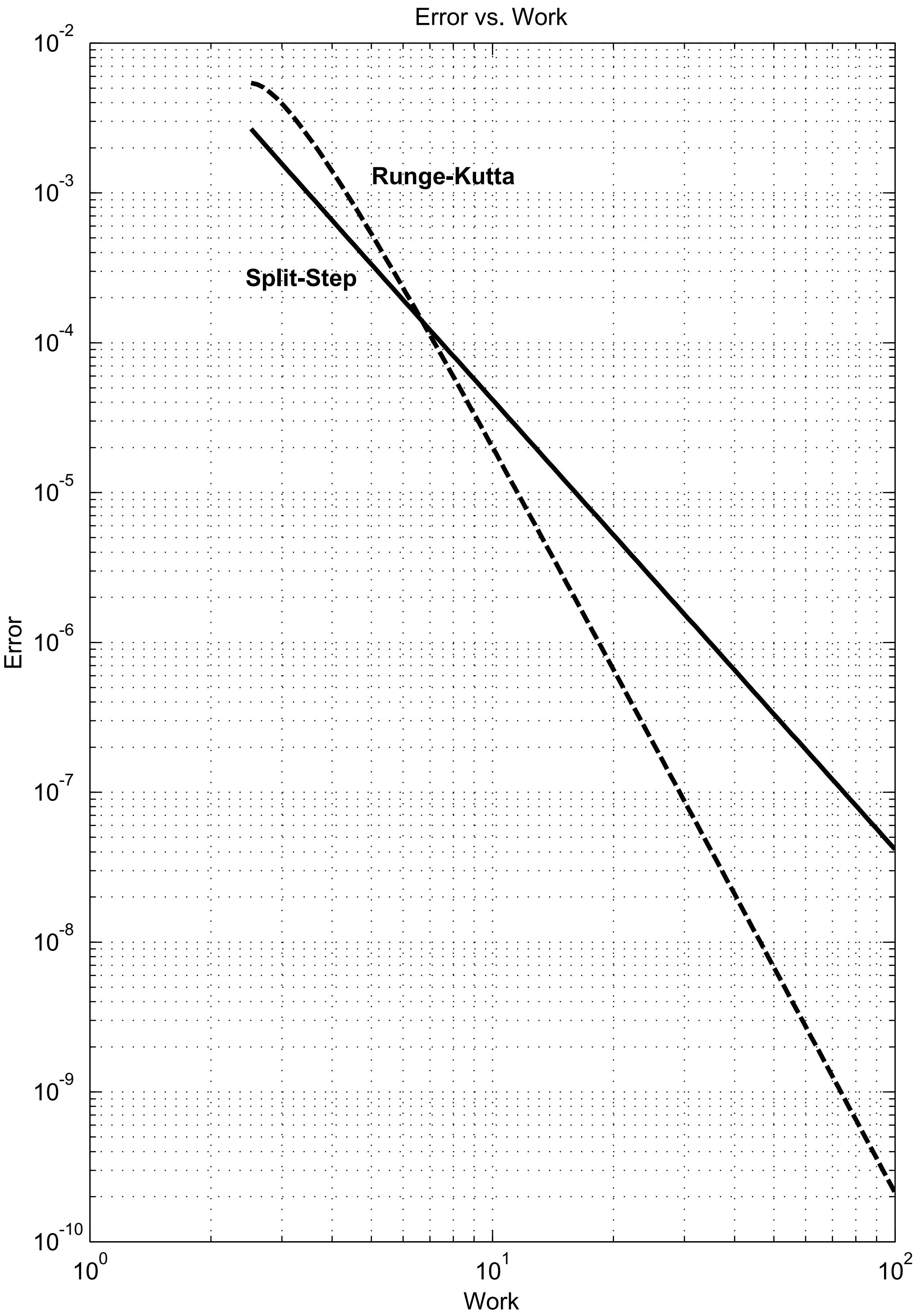
MATLAB code for evaluation of phase errors vs. work for the split-step and Runge-Kutta methods applied to a harmonic oscillator (January 29, 2003 — C. R. Menyuk)

```
% Error_Plot
%
% The phase error for a fixed amount of work is plotted for both the
% split step method and the Runge-Kutta method, applied to the harmonic
% oscillator.

% The basic calculation assumes that since the operator count includes
% one function evaluation per step for the split step method, while it
% includes four function evaluations per step for the Runge-Kutta method,
% the step size for a given amount of work is four times as high for
% the latter method.  It is also inversely proportional to the step size,
% hence we set it equal to the inverse step size for the split step method
% and four times as high for the Runge-Kutta method.

% Input parameters
range = [.4 2];     % the logarithm base 10 of the range of Work
npoints = 101;     % the number of evaluation points

% Set up basic arrays
Inc = (range(2) - range(1))/(npoints - 1);  % increment of the array Work
Work = range(1):Inc:range(2);               % Set up Work array
Work = 10.^Work;
Delta2 = 1.0./Work;                         % Step sizes for split-step
Delta4 = 4.0*Delta2;                        % Step sizes for Runge-Kutta

% Calculate error arrays
err2 = atan(Delta2.*sqrt(1-Delta2.^2/4)./(1-Delta2.^2/2)) - Delta2;
err2 = abs(err2);                           % split step error
err4 = atan(Delta4.*(1-Delta4.^2/6)./(1-(Delta4.^2/2).*(1-Delta4.^2/12))) ...
    - Delta4;      err4 = 0.25*abs(err4);   % Runge-Kutta error

% Plot results on a log-log plot
loglog(Work,[err2;err4])
grid on
```

Output of MATLAB Error Determination


Error vs. Work
Runge-Kutta
Split-Step
Error
Work
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-9
10^-10
10^0
10^1
10^2