Please write your family and given names and **underline** your family name on the front page of your paper.

1. [*10 points*] By an instance of Taylor's theorem, $f(a+h) = f(a) + h\,f'(a) + \frac{h^2}{2}\,f''(a) + \frac{h^3}{3!}\,f'''(\xi_+)$ and a similar one for $f(a-h)$, after some manipulation, we can derive the relation $f'(a) = \frac{f(a+h) - f(a-h)}{2h} + O(h^2)$.
Show the details of this manipulation. Show what is included in the $O(h^2)$ term and indicate any assumptions on f and/or its derivatives.

2.(a) [*12 points*] Consider the function $g(a) = g(a,h) = \frac{\sqrt{a+h} - \sqrt{a-h}}{2h}$ as a function of h, with $0 < a$, and $|h| < a$. Find the condition number $\kappa(h)$ of g and study for what ranges of h it becomes large.

(b) [*12 points*] When $h \to 0$, the computation $g(x) = \frac{\sqrt{x+h} - \sqrt{x-h}}{2h}$ is considered a reasonable approximation to the derivative of $\sqrt{x}$. However, the numerical stability of the calculation of $g(x)$ (as it is given) is problematic for a certain range of h. Explain what problems the computation of the expression $g(x) = \frac{\sqrt{x+h} - \sqrt{x-h}}{2h}$ may give rise to and for what ranges of h. (The answer may be in terms of x and $\varepsilon_{\text{mach}}$.)

(c) [*11 points*] Suggest an alternative (but mathematically equivalent) way of calculating $g(x)$, that does not give rise to the instability problems described in (b).

(d) [*20 points*] Write (a matlab) code that, for a given value of a, and for $h = 10^{-1}, 10^{-2}, \cdots, 10^{-17}$ (i.e. $h = 10^{-i}$, $i = 1, 2, \cdots, 17$), calculates the "exact" value (`d0`) of the derivative of $\sqrt{x}$ at $x = a$, the approximate value (`d1`) by $\frac{\sqrt{a+h} - \sqrt{a-h}}{2h}$ and the approximate value (`d2`) by the alternative formula suggested in (c), and outputs `i, d0, d1, d2, d0-d1, d0-d2` on one line. Plot the errors |d0-d1| and |d0-d2| in log-log scale versus the respective values of h in one plot. Run your code for $a = 1$ and $a = 10$. (So you will have two plots and two sets of output.) Comment on the results.
Notes: Use

```
fprintf('%3d %17.14f %17.14f %17.14f %8.1e %8.1e\n', ...
        i, d0, d1, d2, d0-d1, d0-d2);
```

to format the output.
Save the values of h and of the errors `|d0-d1|` and `|d0-d2|` in respective vectors in order to do the plot.
Use `loglog` for the plot and `axis([1e-18 2e-1 1e-17 1]);` after the plot and before saving/printing the plot. Do not worry if some errors are zero.
Hand-in a copy of your code, output and plot.
You must use double precision. Matlab uses double precision by default. You cannot use any package that adjusts the precision, or any symbolic package.

3. [*20 points*] The series $\sum_{i=1}^{\infty} \frac{1}{i^2}$ are known to converge to $s = \frac{\pi^2}{6}$. A straight calculation of the sum in increasing order of i's up to saturation, shows that the calculated sum approximates the infinite sum within about 8 decimal digits. Suggest a way to calculate the sum with higher precision and explain.
Write (a matlab) code that computes the sum (in increasing order of i's) up to saturation, and output the last i used, the "exact" sum s, the calculated sum (`s1`), and the relative error $\frac{s - s1}{s}$. Write also the code to implement the summation with higher precision, and output the highest i used, the "exact" sum s, the calculated sum (`s2`), and the relative error $\frac{s - s2}{s}$. Because this more accurate way includes more terms, it becomes inefficient, so aim for an accuracy of about 10 decimal digits. Comment on the results. Could you approximately predict the number of terms needed to reach saturation in the straight calculation of the sum? Explain. How does the relative error in the more accurate way behaves as the number of terms included increases?
Use format `fprintf('%12d %18.16f %18.16f %8.1e\n', i-1, s0, s1, (s0-s1)/s0)` and similarly for `s2`.
Hand-in a copy of your code and output.
You must use double precision. Matlab uses double precision by default. You cannot use any package that adjusts the precision, or any symbolic package.

4. [*15 points*] Prove that the product of two tridiagonal $n \times n$ matrices is a pentadiagonal matrix, more precisely, a matrix with semi-bandwidth at most 2. (Your proof must be general, for any n and any entries of the tridiagonal matrices.)