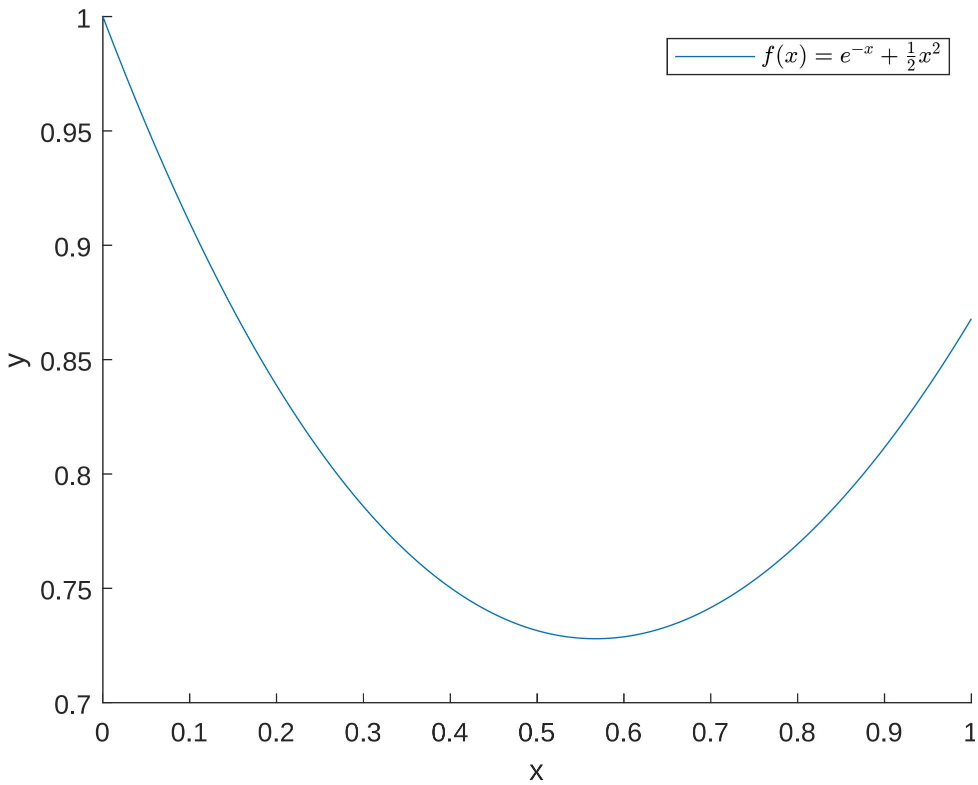
1. (a)
```
f = @(x) exp(-x) + x.^2./2;
x = linspace(0,1,200);
plot(x,f(x));
xlabel('x');
ylabel('y');
leg = legend('$f(x) = e^{-x} + \frac{1}{2}x^2$');
set(leg, 'Interpreter', 'LaTeX')
```

The result is:



There appears to be a minimiser somewhere in the interval $[0.5, 0.6]$.

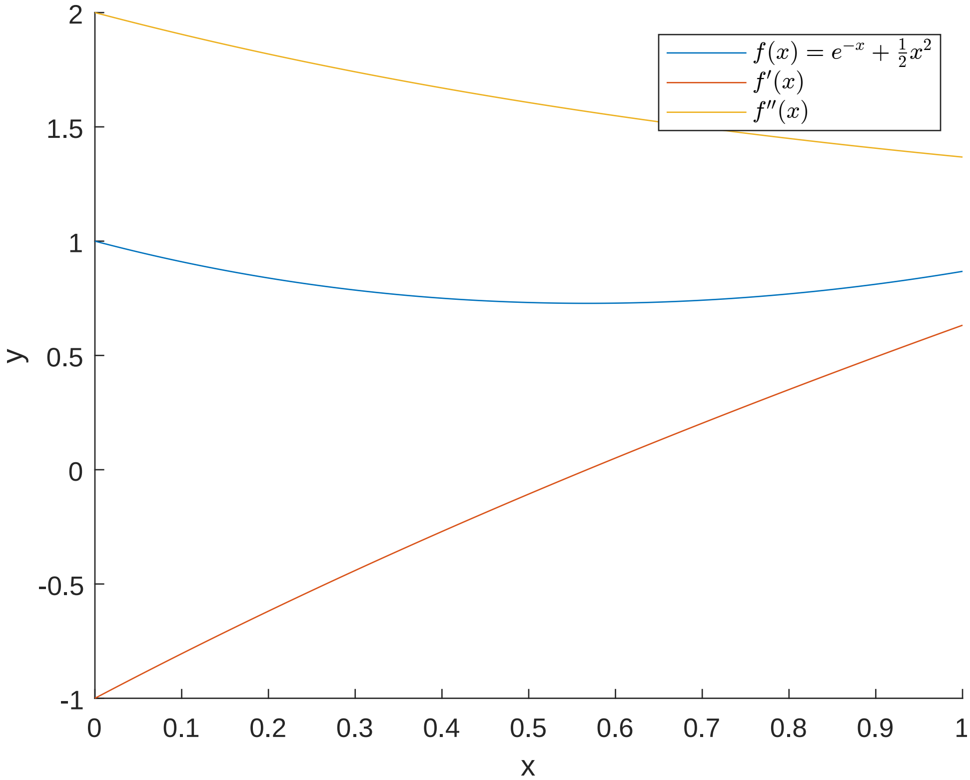
(b) By the FONC, a minimiser $\mathbf{x}^*$ must satisfy $f'(x^*) = 0$. We have

$$f'(x) = 0 \implies -e^{-x} + x = 0,$$

but algebraically solving this using elementary functions is not possible.

(c)
```
f = @(x) exp(-x) + x.^2./2;
fd = @(x) -exp(-x) + x;
fdd = @(x) exp(-x) + 1;
x = linspace(0,1,200);
hold on
plot(x,f(x));
plot(x,fd(x));
plot(x,fdd(x));
xlabel('x');
ylabel('y');
leg = legend('$f(x) = e^{-x} + \frac{1}{2}x^2$','$f''(x)$', '$f''''(x)$');
set(leg, 'Interpreter', 'LaTeX')
```

The plot is shown on the next page.



2. (a) As $a = 0$ and $d = 3$, with $\rho = \frac{3-\sqrt{5}}{2} \approx 0.382$, we have

$$b = a + \rho(d-a) = 0.382 \times 3 = 1.146,$$
$$c = d - \rho(d-a) = 3 - 0.382 \times 3 = 1.854.$$

   (b) From the diagram, $f(c) < f(b)$. So we replace a with b, making the new values $a = 1.146$ and $d = 3$.

3. (a) Here we lay out the iterations in table form.

iteration	a	d	b	c	$f(b)$	$f(c)$	$f(b) < f(c)$?	update
1	0	1	0.382	0.618	0.755	0.730	no	let $a = b$
2	0.382	1	0.618	0.764	0.730	0.0.758	yes	let $d = c$

In the second row, b and c are obtained as follows:

$$b = 0.382 + 0.382 \times (1 - 0.382) = 0.618,$$
$$c = 1 - 0.382 \times (1 - 0.382) = 0.764.$$

So at the end of this iteration we have $a = 0.382$ and $d = 0.764$.

   (b) Note that to find the minimiser within $\pm 10^{-6}$, the length of interval must be no more than 2×10^{-6}.

```
rho = (3 - sqrt(5))/2;
a = 0;
d = 1;
iterations = 0;
while abs(a-d) > 2e-6
    iterations = iterations+1;
    b = a + rho*(d-a);
    c = d - rho*(d-a);
    if f(b) < f(c)
        d = c;
    else
        a = b;
    end
end
% The minimiser is in the interval [a,d], so we take the midpoint.
fprintf('After %d iterations, ', iterations)
fprintf('the minimiser is approximately %.6f.\n', (a+d)/2);
```

After 28 iterations, the minimiser is approximately 0.567143.

(c) After the N-th iteration of the golden section method, the interval has size $(1-\rho)^N \times (d-a)$, where a and d are the initial values. Thus we require

$$(1-\rho)^N \leqslant 2 \times 10^{-6} \iff N \geqslant \log_{1-\rho}(2 \times 10^{-6}) = \frac{\log(2 \times 10^{-6})}{\log(1-\rho)}$$

Writing `log(2e-6)/log(1-rho)` in MATLAB shows that we need at least 27.269 iterations, which is first attained when $N = 28$.

4. One must assume that the function has exactly one maximiser on the initial interval, and then simply invert the comparison: if $f(b) > f(c)$, let $d = c$; otherwise, let $a = b$.

5. The midpoint is 2.5, and from the diagram we can see that at $x = 2.5$ the derivative is negative. So the function is decreasing at $x = 2.5$, implying the minimiser is to the right; hence, the next interval is $[2.5, 5]$.

6. (a) Here we lay out the iterations in table form.

iteration	a	c	b	$f'(b)$	$f(b) < 0$?	update
1	0	1	0.5	-0.107	yes	let $a = b$
2	0.5	1	0.75	0.278	no	let $c = b$

So at the end of this iteration we have $a = 0.5$ and $c = 0.75$.

(b)
```
a = 0;
c = 1;
iterations = 0;
while abs(a-c) >= 2e-6
    iterations = iterations+1;
    b = (a+c)/2;
    if fd(b) < 0
        a = b;
    elseif fd(b) > 0
        c = b;
    else
        a = b;
        c = b;
    end
end
% The minimiser is in the interval [a,c], so we take the midpoint.
fprintf('After %d iterations, ', iterations)
fprintf('the maximiser is approximately %.6f.\n', (a+c)/2);
```

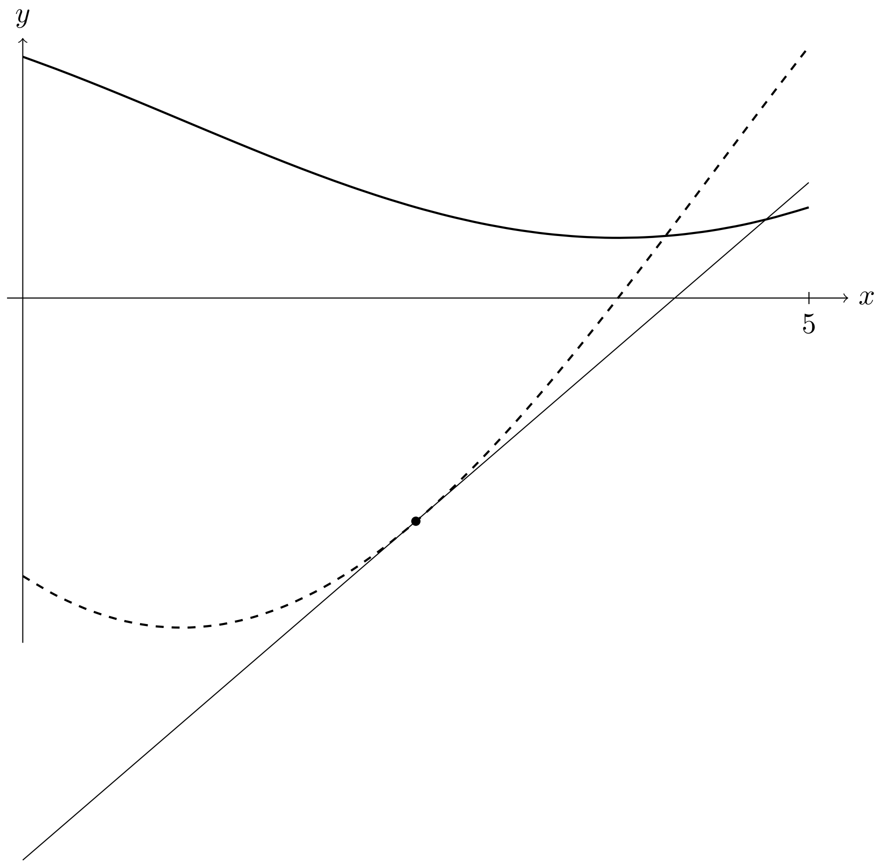
(c) After the N-th iteration of the bisection method, the interval has size $(1/2)^N \times (c-a)$, where a and c are the initial values. Thus we require

$$(1/2)^N \leqslant 2 \times 10^{-6} \iff 2^N \geqslant 0.5 \times 10^6 \iff N \geqslant \log_2(0.5 \times 10^6)$$

Writing `log2(0.5e6)` in MATLAB shows that we need at least 18.932 iterations, which is first attained when $N = 19$.

7. One must assume that the function has exactly one maximiser on the initial interval, and then simply invert the comparisons: if $f'(b) > 0$, let $a = b$; else if $f'(b) < 0$, let $c = b$, else return $[b, b]$.

8. The line tangent to the derivative at $x = 2.5$ is shown below.



It intersects the x-axis at approximately $x = 4$, so the next iteration will have $x \approx 4$.

9. (a) Here we lay out the iterations in table form.

iteration	x	$f'(x)$	$f''(x)$	evaluation
1	1	0.632	1.368	$1 - 0.632/1.368 = 0.538$
2	0.538	-0.046	1.584	$0.538 + 0.046/1.584 = 0.567$

At the end of this iteration, we have $x \approx 0.567$.

(b)
```
x = 1;
x_prev = Inf;
iterations = 0;
while abs(x_prev - x) >= 2e-6
    x_prev = x;
    x = x - fd(x)/fdd(x);
    iterations = iterations+1;
end
fprintf('After %d iterations, ', iterations)
fprintf('the minimiser is approximately %.6f.\n', x);
```

After 4 iterations, the maximiser is approximately 0.567143.

10. The algorithm needs no adjustments in this case; only the assumptions must be modified. The assumption that the function has exactly one minimiser must be replaced with the assumption that there is exactly one maximiser.