Algorithm

The algorithm (of the main function) in pseudocode:

```
Define a subset of C (e.g.  [2;3] × [2i;2i]),
Divide the subset into m × n rectangles (tessellation).
Compute for every rectangle the position of its
center-point in the complex plane.
for k = 1 to m × n
test for every rectangle if its center-point belongs to
the Mandelbrot set.  If yes, print a ∗, if not, print a
blank.  If k == m, print a newline.
end
```

In 1978: m = 80, n = 49. On today terminal emulations like *konsole* one can use larger values if the length of one line is increased. (Use e.g. ctrl+mouse-wheel, depends on your settings.)

Algorithm

To check if a point is in the Mandelbrot set, one has to iterate the formula

$$\begin{aligned} z_{k+1} &= z_k^2 + z_0 \\ z_0 &= c \end{aligned}$$

If z_{k+1} remains bounded, then c belongs to the Mandelbrot set. TU

Algorithm

```
function mandelbrot(c,threshold,N)
  z = c;
  for k = 1 to N
    z = z^2+c;
  end   if abs(z) <= threshold
    return true;
  else
    return false;
  end
end
```

Result


code : bash — Konsole
Datei
Bearbeiten
Ansicht
Lesezeichen
Einstellungen
Hilfe
uwe@uwe-N24-25JU:~/Arbeit/Lehre/Eigene LV/HPC/code$ g++ mandelbrot.c -o mandelbrot
uwe@uwe-N24-25JU:~/Arbeit/Lehre/Eigene LV/HPC/code$ ./mandelbrot
uwe@uwe-N24-25JU:~/Arbeit/Lehre/Eigene LV/HPC/code$
code : bash