```
m = (y1-y0) / (x1-x0)
x = x0
y = y0
while (x <= x1) {
        setPixel(x, round(y))              // round(y) = floor(y + 0.5)
        x = x + 1
        y = y + m
}
```

Convert m and y to fractions:

m = m_num / m_den, and

y = y_int + y_num / y_den

```
m_num = y1 – y0
m_den = x1 – x0
x = x0
y_int = y0
y_num = 0
y_den = x1 – x0
while (x <= x1) {
        setPixel(x, floor( (y_int + y_num / y_den) + 0.5))
        x = x + 1
        y_num = y_num + m_num
        if (y_num >= y_den) {                    // fraction bigger than 1
                y_num = y_num – y_den
                y_int = y_int + 1
        }
}
```

If we let (y_int, y_num, y_den) store y + 0.5 rather than y, then the floor becomes

floor((y_int + y_num / y_den))

which equals y_int because we fix the fraction if it overflows. To store y + 0.5, we need an even denominator, so we multiply it (and the numerator) by 2. We do this to m_num and m_den as well, to make the addition of y and m easier.

```
m_num = 2(y1 – y0)
m_den = 2(x1 – x0)
x = x0
y_int = y0
y_num = x1 – x0
y_den = 2(x1 – x0)
while (x <= x1) {
        setPixel(x, y_int)
        x ++
        y_num += m_num
        if (y_num – y_den >= 0) {
                y_num –= y_den
                y_int ++
        }
}
```

Now we move the statement y_num += m_num inside the if statement. To do this, we add the else clause:

```
        y_num += m_num
        if (y_num – y_den >= 0) {
                y_num –= y_den
                y_int ++
        }
        else {
        }
```

then move the statement into both branches, adjusting the condition as we do:

```
        if (y_num – y_den + m_num >= 0) {
                y_num += m_num
                y_num –= y_den
                y_int ++
        }
        else {
                y_num += m_num
        }
```

```
m_num = 2(y1 – y0)
m_den = 2(x1 – x0)
x = x0
y_int = y0
y_num = x1 – x0
y_den = 2(x1 – x0)
while (x <= x1) {
        setPixel(x, y_int)
        x ++
        if (y_num – y_den + m_num >= 0) {
                y_num += m_num – y_den
                y_int ++
        }
        else {
                y_num += m_num
        }
}
```

To simplify the if statement's condition, we'll define

$$err = y\_num - y\_den + m\_num$$

then wherever y_num was updated, we update err accordingly.

At initialization,

$$\begin{aligned} err &= y\_num - y\_den + m\_num \\ &= (x1 - x0) - 2(x1 - x0) + 2(y1 - y0) \\ &= 2(y1 - y0) - (x1 - x0) \end{aligned}$$

We also define

$$k = m\_num - y\_den = 2(y1 - y0) - 2(x1 - x0)$$

```
m_num = 2(y1 – y0)
m_den = 2(x1 – x0)
x = x0
y_int = y0
err = 2(y1 – y0) – (x1 – x0)
y_den = 2(x1 – x0)
k = 2(y1 – y0) – 2(x1 – x0)
while (x <= x1) {
        setPixel(x, y_int)
        x ++
        if (err >= 0) {
                err += k
                y_int ++
        }
        else {
                err += m_num
        }
}
```

Remove the unused variables m_den and y_den. Rename y_int to y.

```
m_num = 2(y1 – y0)
x = x0
y = y0
err = 2(y1 – y0) – (x1 – x0)
k = 2(y1 – y0) – 2(x1 – x0)
while (x <= x1) {
        setPixel(x, y)
        x ++
        if (err >= 0) {
                err += k
                y ++
        }
        else {
                err += m_num
        }
}
```