Computing Polynomials

Alexandre David
B2-206

The Problem

- How to compute efficiently polynomials of the form

$$\sum_{i=0}^{n} a_i x^i$$

- Naïve approach: Compute each term and sum-up. If naïve power algorithm is used: $n(n+1)/2$ multiplications + n additions.
- Better? ?

Idea

- Rewrite the polynomial as:

$$\sum_{i=0}^{n} a_i x^i = a_0 + x(a_1 + x(a_2 + \ldots x(a_n)\ldots))$$

Horner's rule

- Compute "a" as:

  a = a_n

  a = a_{n-1} + x*a

  ...

  a = a_1 + x*a

  a = a_0 + x*a

Algorithm

```
a = ai[n]
while (n>0) do
        n = n-1
        a = ai[n] + x*a
done
return a
```

Notes:
- Efficient if most ai[n] are not null.
- Not necessary the best precision.

- Optimized for some DSPs that have a *multiply and accumulate* instruction.