

[Accelerate](#) / [vDSP](#) / `evaluatePolynomial(usingCoefficients:withVariables:result:)`

Type Method

evaluatePolynomial(usingCoefficients:withVariables:result:)

Evaluates a double-precision polynomial using specified coefficients and variables.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst | macOS 10.15+ | tvOS 13.0+ | visionOS | watchOS 6.0+

```
static func evaluatePolynomial<U, V>(
    usingCoefficients coefficients: [Double],
    withVariables variables: U,
    result: inout V
) where U : AccelerateBuffer, V : AccelerateMutable
Buffer, U.Element == Double, V.Element == Double
```

Parameters

`coefficients`

An array that contains the coefficients.

`variables`

An array that contains the independent variables.

`result`

An array that receives the result of the calculation.

Discussion

For example, the following code evaluates the polynomial with the coefficients [10.0, 20.0, 30.0] and the variables [7.0, 5.0]:

```
let coefficients: [Double] = [10, 20, 30]
let variables: [Double] = [7, 5]

let result = [Double](
    unsafeUninitializedCapacity: variables.count) {
    buffer, initializedCount in

    vDSP.evaluatePolynomial(usingCoefficients: coefficients,
                            withVariables: variables,
                            result: &buffer)

    initializedCount = 2
}

// Prints "[660.0, 380.0]".
//    result[0] = (10 * 72) + (20 * 71) + (30 * 70) = 660
//    result[1] = (10 * 52) + (20 * 51) + (30 * 50) = 380
print(result)
```

See Also

Related Documentation

vDSP_vpoly

Evaluates a single-precision polynomial using specified coefficients, variables, and strides.

Double-precision polynomial evaluation

```
static func evaluatePolynomial<U>(usingCoefficients: [Double], with
Variables: U) -> [Double]
```

Returns a double-precision evaluated polynomial using specified coefficients and variables.