

```

/*=====*\
filename: rect_pol.mu
-----
scopo: routines per la conversione tra coordinate
       rettangolari e polari e viceversa
-----
autore: Claudio Marsan
        Liceo cantonale di Mendrisio
        Via Agostino Maspoli
        CH-6850 Mendrisio
        claudio.marsan@liceomendrisio.ch
-----
ultima modifica: 28.08.2005
-----
testato con: MuPAD Pro 3.1.1 for Windows
-----
OS: Microsoft Windows XP Professional sp2
-----
uso: read("../\rect_pol.mu"):
/*=====*/

```

```

// Converta le coordinate polari (r, theta) nelle coordinate
// rettangolari (x, y); theta è in radianti.

```

```
P2R := (r, theta) -> [r*cos(theta), r*sin(theta)]:
```

```

// Converta le coordinate polari (r, theta) nelle coordinate
// rettangolari (x, y); theta è in gradi sessadecimali.

```

```
P2Rdeg := (r, theta) -> [r*cos(theta*PI/180), r*sin(theta*PI/180)]:
```

```

// Converta le coordinate rettangolari (x, y) nelle coordinate
// polari (r, theta); theta è in radianti.

```

```
R2P := proc(x, y)
```

```
  local X, Y, r, theta;
```

```
begin
```

```
  X := float(x);
```

```
  Y := float(y);
```

```
  r := sqrt(x^2 + y^2);
```

```
  if (float(r) = 0.0) then
```

```
    return([0, 0]);
```

```
  end_if;
```

```
  if (X <> 0.0) then
```

```
    theta := arctan(y/x);
```

```
    if (X > 0.0) and (Y >= 0.0) then // I quadrante
```

```
      return([r, theta]);
```

```
    end_if;
```

```
    if (X < 0.0) and (Y >= 0.0) then // II quadrante
```

```
      return([r, theta + PI]);
```

```
    end_if;
```

```
    if (X < 0.0) and (Y < 0.0) then // III quadrante
```

```
      return([r, theta + PI]);
```

```
    end_if;
```

```
    if (X > 0.0) and (Y < 0.0) then // IV quadrante
```

```
      return([r, theta + 2*PI]);
```

```
    end_if;
```

```
  else
```

```
    if (Y >= 0.0) then
```

```
      return ([r, PI/2]);
```

```
    else
```

```
      return ([r, 3*PI/2]);
```

```
    end_if;
```

```
  end_if;
```

```
end:
```

```

// Converta le coordinate rettangolari (x, y) nelle coordinate
// polari (r, theta); theta è in gradi sessadecimali.

```

```
R2Pdeg := (x, y) -> [R2P(x, y)[1], R2P(x, y)[2]*180/PI]:
```

```
// MC1, 28.08.2005
```

