

```

/*=====*\
filename: rect_sph.mu
-----
scopo: routines per la conversione tra coordinate
       rettangolari e sferiche e viceversa
-----
autore: Claudio Marsan
        Liceo cantonale di Mendrisio
        Via Agostino Maspoli
        CH-6850 Mendrisio
        claudio.marsan@liceomendrisio.ch
-----
ultima modifica: 11.10.2005
-----
testato con: MuPAD Pro 3.1.1 for Windows
-----
OS: Microsoft Windows XP Professional sp2
-----
uso: read("../\rect_sph.mu"):
/*=====*/

```

```

// Converta le coordinate sferiche (r, theta, phi) nelle coordinate
// rettangolari (x, y, z); theta e phi sono in radianti.

```

```

S2R := proc(r, theta, phi)
  local r0, theta0, phi0;
begin
  r0 := float(r);
  theta0 := float(theta);
  phi0 := float(phi);
  if (r0 < 0.0) then
    error("Il raggio vettore è negativo.");
  end_if;
  if ((theta0 < 0.0) or (theta0 > float(PI))) then
    error("La colatitudine non è compresa in [0, 2*PI].");
  end_if;
  if ((phi0 < 0.0) or (phi0 >= 2*float(PI))) then
    error("La longitudine non è compresa in [0, 2*PI[.");
  end_if;
  return ([r*sin(theta)*cos(phi),
          r*sin(theta)*sin(phi),
          r*cos(theta)]);
end:

```

```

// Converta le coordinate sferiche (r, theta, phi) nelle coordinate
// rettangolari (x, y, z); theta e phi sono in gradi sessadecimali.

```

```

S2Rdeg := proc(r, theta, phi)
  local r0, theta0, phi0;
begin
  r0 := float(r);
  theta0 := float(theta*PI/180);
  phi0 := float(phi*PI/180);
  if (r0 < 0.0) then
    error("Il raggio vettore è negativo.");
  end_if;
  if ((theta0 < 0.0) or (theta0 > float(PI))) then
    error("La colatitudine non è compresa in [0, 180°].");
  end_if;
  if ((phi0 < 0.0) or (phi0 >= 2*float(PI))) then
    error("La longitudine non è compresa in [0, 360°[.");
  end_if;
  return ([r*sin(theta*PI/180)*cos(phi*PI/180),
          r*sin(theta*PI/180)*sin(phi*PI/180),
          r*cos(theta*PI/180)]);
end:

```

```

// Converta le coordinate rettangolari (x, y, z) nelle coordinate
// sferiche (r, theta, phi); theta e phi sono in radianti.

```

```

R2S := proc(x, y, z)
  local r, x0, y0, theta, phi;

```

```
begin
  r := sqrt(x^2 + y^2 + z^2);
  if (float(r) = 0.0) then
    return([0, 0, 0]);
  else
    theta := arccos(z/r);
    x0 := float(x);
    y0 := float(y);
    if (x0 = 0.0) then
      if (y0 > 0.0) then
        phi := PI/2;
      elif (y0 < 0.0) then
        phi := 3*PI/2;
      else
        phi := 0;
      end_if;
    else
      phi := arctan(y/x);
    end_if;
    if (x0 < 0.0) then
      phi := phi + PI;
    elif ((x0 > 0.0) and (y0 < 0.0)) then
      phi := phi + 2*PI;
    end_if;
    return([r, theta, phi]);
  end_if;
end:

// Convert le coordinate rettangolari (x, y, z) nelle coordinate
// sferiche (r, theta, phi); theta e phi sono in gradi sessadecimali.
R2Sdeg := proc(x, y, z)
  local P;
begin
  P := R2S(x, y, z);
  return([P[1], float(P[2]*180/PI), float(P[3]*180/PI)]);
end:
```