Teorema divergentei:

Una din metodele de calcul al integralelor de flux 3D este utilizarea teoremei divergentei lui Gauss care permite trecerea de la o integrala de suprafata la o integrala tripla a divergentei.

$$\oiint \vec{a} d\vec{S} = \int \operatorname{div} \vec{a} \, dV$$

Cand divergenta unui camp vectorial este constanta, inmultim pur si simplu divergenta cu volumul inconjurat de suprafata inchisa.

In exemplul dat campul vectorial este F(x,y,z)=(2x,3y,-z) iar suprafata este cea a unei sfere de raza R=2

```
Clear[F, x, y, z, DiverF, r, Vol]
Needs["Calculus`VectorAnalysis`"]
F[x_, y_, z_] = {2 x, 3 y, -z};
DiverF[x_, y_, z_] = Div[F[x, y, z], Cartesian[x, y, z]]
(* Volume of a sphere 4/3 π r^3 *)
r = 2;
Vol = 4/3 * π * r^3;
DiverF[x, y, z] * Vol
```

4

$$\frac{128 \pi}{3}$$

Calculul integralei de flux peste o suprafata neteda inchisa in spatiul tridimensional poate fi facut cu ajutorul integralei triple a divergentei campului ,peste volumul marginit de suprafata.

Fie campul vectorial f(x,y,z)=(x-y, y-4xz, z) si o suprafata de forma unei cutii cu dimensiunile (0,4)×(0,2)×(0,3):

```
Clear[F, x, y, z, DiverF]
Needs["Calculus`VectorAnalysis`"]
F[x_, y_, z_] = {x - y, y - 4 * x * z, x * z};
DiverF[x_, y_, z_] = Div[F[x, y, z], Cartesian[x, y, z]]
∫_0^4 ∫_0^2 ∫_0^3 DiverF[x, y, z] dz dy dx
```

2 + x

96

Teorema lui Stokes:

Teorema trecerii de la o integrala de suprafata la una de contur:

$$\iint \operatorname{curl} \vec{a} d\vec{S} = \oint \vec{a} d\vec{l}$$

a) Sa calculam mai intai integrala de suprafata $\int_{\square}^{\square} \int_{S}^{\square} \operatorname{Curl}(F).N \, d\sigma$ peste paraboloidul z = x^2 + y^2 pentru x^2 + y^2 <= 9.

```
Clear[F, S, r, the, CuF, Nhat, x, y, z]
Needs["Calculus`VectorAnalysis`"]
(* definim campul vectorial F si suprafata S *)
F[x_, y_, z_] = {x*y, y*z, x*z}
S[x_, y_] = x^2 + y^2;
(* calculam vectorul normal la suprafata *)
Norm[x_, y_, z_] = {D[S[x, y], x], D[S[x, y], y], -1} / Sqrt[1 + D[S[x, y], x]^2 + D[S[x, y], y]^2]
(* calculam rotationalul campului vectorial in cartezian *)
CuF[x_, y_, z_] = Curl[F[x, y, z], Cartesian[x, y, z]]
Sx[x_, y_] = D[S[x, y], x]^2; (* Sx = (∂x S)^2 *)
Sy[x_, y_] = D[S[x, y], y]^2; (* Sy = (∂y S)^2 *)
(* asiguram trecerea la coordonate cilindrice *)
∫_0^(2π) ∫_0^3 (CuF[r*Cos[θ], r*Sin[θ], z].Norm[r*Cos[θ], r*Sin[θ], z]) *
   Sqrt[1 + Sx[r*Cos[θ], r*Sin[θ]] + Sy[r*Cos[θ], r*Sin[θ]]] * r dr dθ
```

$\{x\,y,\ y\,z,\ x\,z\}$

$$\left\{\frac{2\,x}{\sqrt{1+4\,x^2+4\,y^2}},\ \frac{2\,y}{\sqrt{1+4\,x^2+4\,y^2}},\ -\frac{1}{\sqrt{1+4\,x^2+4\,y^2}}\right\}$$

$\{-y,\ -z,\ -x\}$

0

b) Calculam integrala de linie de tipul $\int_C F.dR$ unde R(t)=(Cos(t), 3Sin(t), 4)

```
Clear[R, F, x, y, z, t, ds]
x[t_] = Cos[t];
y[t_] = Sin[t];
z[t_] = 4;
R[t_] = {x[t], y[t], z[t]};
F[t_] = {x[t]*y[t], y[t]*z[t], x[t]*z[t]}
∫_0^(2π) F[t].R'[t] dt
```

{Cos[t] Sin[t], 4 Sin[t], 4 Cos[t]}

0

Operatori diferentiali in coordonate carteziene, cilindrice si sferice:

Elemente de baza:
Facem apel la pachetul de programe specifice analizei vectoriale
<<Calculus`VectorAnalysis`
Pachetul permite calculul operatorilor diferentiali Grad, Div, Rotor (Curl) si Laplacian in orice fel de sisteme de coordonate
Este un pachet des utilizat in aplicatii din mecanica analitica, electromagnetism, electrodinamica, mecanica cuantica etc.
Stabilim prin comanda SetCoordinates[TipCoord[var1,var2,var3]] tipul de coordonate dorit si variabilele specifice fiecaruia

```
DeclarePackage["Calculus`VectorAnalysis`", {"Div", "Grad", "Curl"}]
(*<<Calculus`VectorAnalysis` *)

SetCoordinates[Cartesian[x, y, z]]
```

Cartesian[x, y, z]

```
Div[{x, y^2, x}]
```

$1 + 2\,y$

```
Grad[x^2 + y^2, Cartesian[x, y, z]]
```

{2 x, 2 y, 0}

```
Curl[{x, y^2, x}]
```

{0, -1, 0}

```
Laplacian[x^2 + y^2]
```

4

```
SetCoordinates[Spherical[r, theta, phi]]
```

```
General::spell1 : Possible spelling error:
   new symbol name "theta" is similar to existing symbol "Ttheta".
```

Spherical[r, theta, phi]

```
Grad[r^2 Sin[theta]]
```

{2 r Sin[theta], r Cos[theta], 0}

```
Div[{r^2 * theta, r * Sin[theta] - r^2, r}]
```

$$\frac{1}{r^2}\left(\mathrm{Csc}[\mathrm{theta}]\left(4\,r^3\,\mathrm{theta}\,\mathrm{Sin}[\mathrm{theta}] + r^2\,\mathrm{Cos}[\mathrm{theta}]\,\mathrm{Sin}[\mathrm{theta}] + r\,\mathrm{Cos}[\mathrm{theta}]\left(-r^2 + r\,\mathrm{Sin}[\mathrm{theta}]\right)\right)\right)$$

```
Curl[{r^2 * theta, r * Sin[theta] - r^2, r}]
```

$$\left\{\mathrm{Cot}[\mathrm{theta}],\ -2,\ \frac{-2\,r^2 + r\,\mathrm{Sin}[\mathrm{theta}] + r\,(-2\,r + \mathrm{Sin}[\mathrm{theta}])}{r}\right\}$$

```
Laplacian[r^2 Sin[theta]]
```

$$\frac{\mathrm{Csc}[\mathrm{theta}]\left(r^2\,\mathrm{Cos}[\mathrm{theta}]^2 + 5\,r^2\,\mathrm{Sin}[\mathrm{theta}]^2\right)}{r^2}$$

```
SetCoordinates[Cylindrical[r, theta, z]]
```

Cylindrical[r, theta, z]

```
Grad[r^2 - 2 * r * Sin[theta] + z^2]
```

{2 r - 2 Sin[theta], -2 Cos[theta], 2 z}

```
Div[{r^2, 2 * r * Sin[theta], z^2}]
```

$$\frac{3\,r^2 + 2\,r\,z + 2\,r\,\mathrm{Cos}[\mathrm{theta}]}{r}$$

```
Curl[{r^2, 2 * r * Sin[theta], z^2}]
```

{0, 0, 4 Sin[theta]}

```
Laplacian[r^2 - 2 * r * Sin[theta] + z^2]
```

6

Campul electric generat de sarcini punctuale

Inainte de a utiliza acest exemplu este necesara o discutie despre notinea de "model " in *Mathematica,* datorita frecventei aparitiei necesitatii "obiectelor model". Un obiect de genul x_, poate fi orice expresie care implicit va avea numele x, nume care poate fi folosit in partea dreapta a unei reguli de transformare. Cand utilizati x_, toate aparitiile de spatii cu numele x din expresiile utilizate vor folosi aceeasi expresie:
f[x_ , x_] se va utiliza numai in expresiile in care cele doua argumente ale lui f sunt aceleasi
f[_ , _] se va utiliza numai in expresiile de forma f[x,y] unde x ≠y.
Spre exemplu:

```
{f[x, x], f[x, y]} /. f[x_, x_] → g[x]
```

```
{g[x], f[x, y]}
```

x : pattern inseamna model care este atribuit cu numele x
Spre exemplu:

```
f[x1^y1] /. f[model : _^_] → g[model]
```

```
g[x1^y1]
```

```
f[x1^y1] /. f[model : _^n_] → g[model, n]
```

```
g[x1^y1, y1]
```

```
(* esponentul este numit n iar intreg obiectul este x*)
```

```
{f[g[x1], g[x1]], f[g[x1], g[x2]]} /. f[model : g[_], model_] → ω[model]
```

```
{ω[g[x1]], f[g[x1], g[x2]]}
```

```
Needs["Graphics`PlotField`"];
Needs["Graphics`PlotField3D`"];
```

```
Clear["Global`*"];
```

```
Sarc[q_, r0_: {0, 0, 0}, r_: {x, y, z}] := q / Sqrt[(r0 - r).(r0 - r)]
```

```
gr1[distribution_] := PlotGradientField[Evaluate[-distribution /. {z → 0}],
   {x, -2, 2.01}, {y, -2, 2.01}, Graphics`PlotField`ScaleFunction → (1 &)];
```

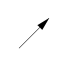
```
gr2[distribution_] :=
  PlotGradientField3D[Evaluate[-distribution], {x, -2, 2.01}, {y, -2, 2.01}, {z, -2, 2.01},
   PlotPoints → 8, Graphics`PlotField3D`ScaleFunction → (1 &), VectorHeads → True];
```

```
Protect[{gr1, gr2, Sarc}];
Off[Clear::"wrsym"];
```

```
Clear["Global`*"];
```

Functia Sarc[q,{x,y,z}] plasează o singură sarcină q în punctul {x,y,z}.
Spre exemplu: **Sarc[+1,{1,0,0}]**
Sã vizualizãm cãmpul produs de o singurã sarcinã în punctul origine {x,y,z}={0,0,0}.
Functia **gr1** va desena în mod automat vectoriul cãmp electric în planul {x,y}id. {-2,2}.

```
gr1[Sarc[+1, {0, 0, 0}]];
```



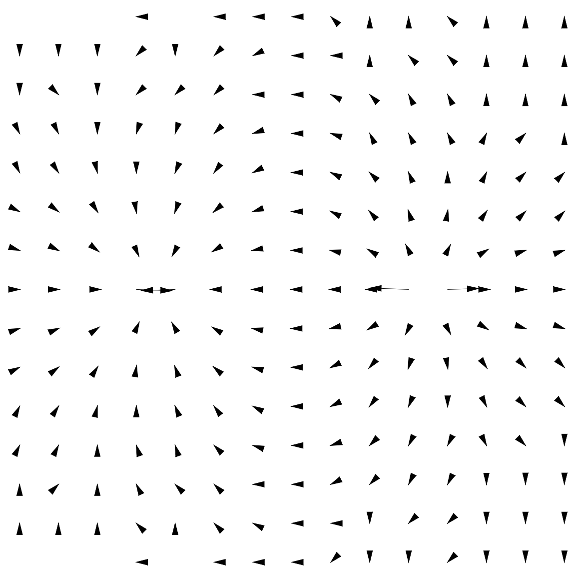
Putem creea un dipol din douã functii **`Sarc`** (dar de semn opus):

```
dip = Sarc[+1, {1, 0, 0}] + Sarc[-1, {-1, 0, 0}]
```

$$-\frac{1}{\sqrt{(-1-x)^2+y^2+z^2}}+\frac{1}{\sqrt{(1-x)^2+y^2+z^2}}$$

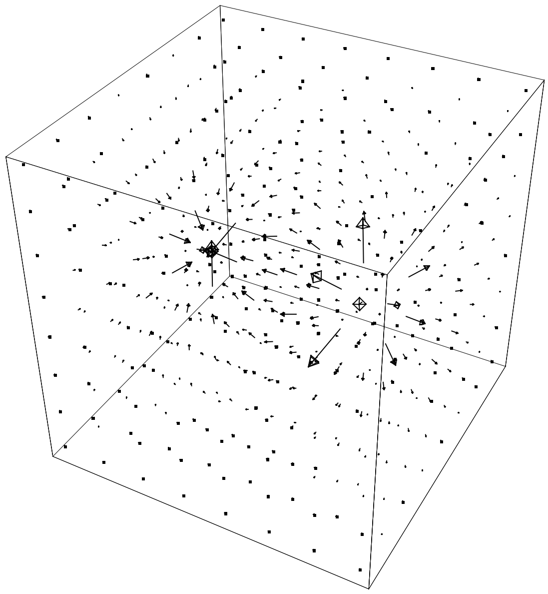
Sã vizualizãm câmpul creat de dipol:

```
gr1[dip];
```



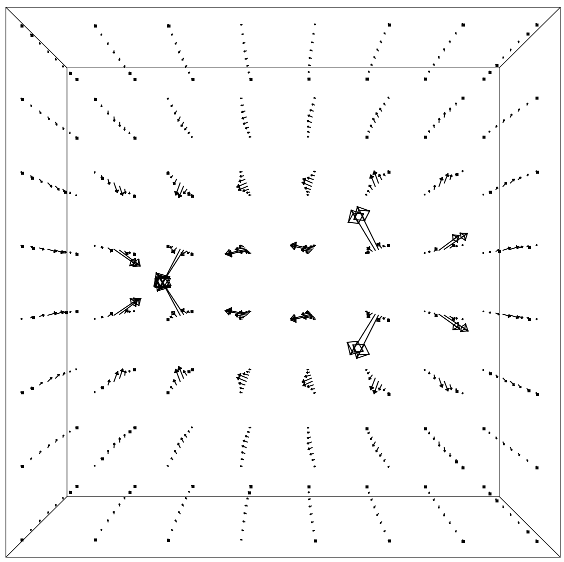
Pentru reprezentarea tridimensionalã a câmpului electric folosim functia **`gr2`**

```
Tridim = gr2[dip];
```



Imaginea rotitã:

```
Show[Tridim, ViewPoint → {0.000, 0.000, 4.000}];
```

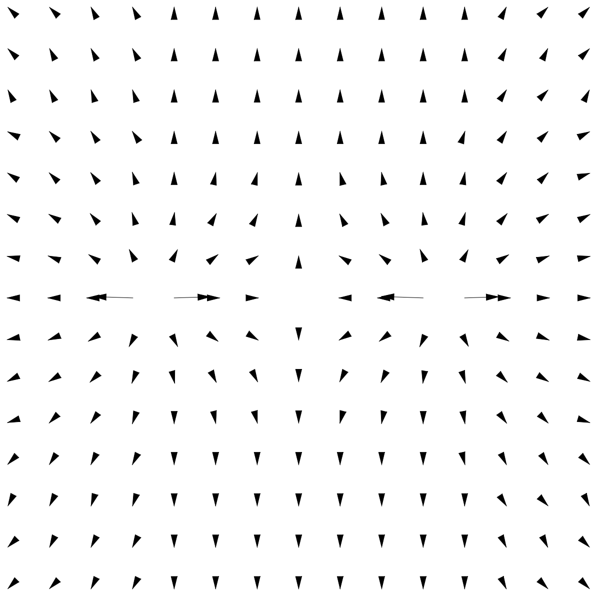


Sã parcurgem acelasi traseu dar cu doua sarcini de acelasi semn!

```
DouaSarc = Sarc[+1, {1, 0, 0}] + Sarc[+1, {-1, 0, 0}]
```

$$\frac{1}{\sqrt{(-1-x)^2+y^2+z^2}}+\frac{1}{\sqrt{(1-x)^2+y^2+z^2}}$$

```
gr1[DouaSarc];
```



- **Incercati pentru trei sarcini plasate in punctele în {1,0,0}, {-1,0,0}, {0,1,0}.**