N-Particle Pairwise Interaction

From a MathGroup Question [mg70316] 12 Oct 2006

Renan Cabrera
cabrer7@uwindsor.ca

```
In[1]:= Needs["TensorCalculus4`Tensorial`"]

In[2]:= DeclareBaseIndices[{1, 2, 3}]

In[3]:= DefineTensorShortcuts[{x, 1}, {{x, r, η, δ}, 2}]

In[4]:= DeclareIndexFlavor[{red, Red}]

In[5]:= SetTensorValues[ηdd[i, j], IdentityMatrix[3]]
        SetTensorValues[ηuu[i, j], IdentityMatrix[3]]

        SetTensorValues[δud[red@i, red@j], IdentityMatrix[3]]
        SetTensorValues[δud[i, j], IdentityMatrix[3]]
```

We can define the coordinates with two indices, the first index for the particle and the second for the component xyz

```
In[9]:= xuu[red@i, j]
```

Out[9]= $x^{i\,j}$

In this case we could define the following metric $\eta_{i\,j}$ to calculate the square distance between two different particles {μ, ν}

```
In[10]:= (xuu[red@μ, i] - xuu[red@ν, i]) (xuu[red@μ, j] - xuu[red@ν, j]) ηdd[i, j]
         % // EinsteinSum[] // FullSimplify
```

Out[10]= $(x^{\mu\,i} - x^{\nu\,i})\,(x^{\mu\,j} - x^{\nu\,j})\,\eta_{i\,j}$

Out[11]= $(x^{\mu\,1} - x^{\nu\,1})^2 + (x^{\mu\,2} - x^{\nu\,2})^2 + (x^{\mu\,3} - x^{\nu\,3})^2$

Defining the metric to be constant

```
In[12]:= PartialD[_][ηdd[_, _], _] = 0
```

Out[12]= 0

so the distance between two particles would be

```
In[13]:= DistanceRule = ruu[red@μ, red@ν] →
           Sqrt[(xuu[red@μ, i] - xuu[red@ν, i]) (xuu[red@μ, j] - xuu[red@ν, j]) ηdd[i, j]]
```

Out[13]= $r^{\mu\nu} \to \sqrt{(x^{\mu i} - x^{\nu i})\,(x^{\mu j} - x^{\nu j})\,\eta_{ij}}$

but now the two subindices of $r^{\mu\nu}$ are understood as particle indices. The coordinate partial derivative of the distance between two particles is

```
In[14]:= PartialD[{x, δ, ν, Γ}][
          xuu[red@σ, k] , xuu[red@ν, m]
         ]
```

Out[14]= $\frac{\partial x^{\sigma k}}{\partial x^{\nu m}}$

Defining a personalized function to simplify this expression in terms of Kronecker deltas

```
In[15]:= PartialDToKronecker[expr_] := Module[{},
           expr /. HoldPattern[PartialD[{x, δ, η, Γ}][
               Tensor[x, {σ_, k_}, {Void, Void}] , Tensor[x, {ν_, m_}, {Void, Void}]]] :→
             Tensor[δ, {σ, Void}, {Void, ν}] Tensor[δ, {k, Void}, {Void, m}]
          ]
```

```
In[16]:= PartialD[{x, δ, η, Γ}][
          xuu[red@σ, k] , xuu[red@ν, m]
         ]
         % // PartialDToKronecker
```

Out[16]= $\frac{\partial x^{\sigma k}}{\partial x^{\nu m}}$

Out[17]= $\delta^{k}{}_{m}\,\delta^{\sigma}{}_{\nu}$

The partial derivative of the distance between particles is calculated as

```
In[18]:= PartialD[{x, δ, η, Γ }][
          ruu[red@μ, red@ν] , xuu[red@σ, k]
         ]
         % /. DistanceRule // Simplify // PartialDToKronecker

         PartialDDistanceRule =
          (PartialD[{x, δ, η, Γ }][ruu[red@μ_, red@ν_] , xuu[red@σ_, k_] ] →
            Simplify[Numerator[%] // Expand // MetricSimplify[η] // KroneckerAbsorb[δ]] /
              ruu[red@μ, red@ν] / 2)
```

Out[18]= $\dfrac{\partial r^{\mu\,\nu}}{\partial x^{\sigma\,k}}$

Out[19]= $\dfrac{((x^{\mu\,j} - x^{\nu\,j})\,(\delta^{i}{}_{k}\,\delta^{\mu}{}_{\sigma} - \delta^{i}{}_{k}\,\delta^{\nu}{}_{\sigma}) + (x^{\mu\,i} - x^{\nu\,i})\,(\delta^{j}{}_{k}\,\delta^{\mu}{}_{\sigma} - \delta^{j}{}_{k}\,\delta^{\nu}{}_{\sigma}))\,\eta_{i\,j}}{2\sqrt{(x^{\mu\,i} - x^{\nu\,i})\,(x^{\mu\,j} - x^{\nu\,j})\,\eta_{i\,j}}}$

Out[20]= $\dfrac{\partial r^{\mu\_\,\nu\_}}{\partial x^{\sigma\_\,k\_}} \to \dfrac{(x^{\mu}{}_{k} - x^{\nu}{}_{k})\,(\delta^{\mu}{}_{\sigma} - \delta^{\nu}{}_{\sigma})}{r^{\mu\,\nu}}$

But our expression is not strictly in tensor notation so we must redefine the rule as follows

```
In[21]:= PartialDDistanceRule = PartialD[{x, δ, η, Γ }][ruu[μ_, ν_] , xuu[σ_, k_] ] →
           (δud[μ, σ] - δud[ν, σ]) (xuu[μ, k] - xuu[ν, k]) / ruu[μ, ν]
```

Out[21]= $\dfrac{\partial r^{\mu\_\,\nu\_}}{\partial x^{\sigma\_\,k\_}} \to \dfrac{(x^{\mu\,k} - x^{\nu\,k})\,(\delta^{\mu}{}_{\sigma} - \delta^{\nu}{}_{\sigma})}{r^{\mu\,\nu}}$

The higher order derivatives are calculated recursively

```
In[22]:= HigherOrderPartialDDistanceRule = ( PartialD[{x, δ, η, Γ }][ ruu[μ_, ν_] , dd_ ] :→
             Fold[
              PartialDToKronecker[PartialD[{x, δ, η, Γ }][#1 , #2 ] /. PartialDDistanceRule] &,
              ruu[μ, ν], dd
             ]);
```

The potential energy is the sum over all the possible particles of

```
In[23]:= U[ruu[red@μ, red@ν]]
```

Out[23]= $U[r^{\mu\,\nu}]$

It is not convenient to put this expression inside a sum expressions so we follow the convention that there is an implicit sum in the indices **{μ, ν}** for all the interactions. The expressions involved cannot be cast into a strict tensor format with balanced indices so we are forced to use Table and Sum instead of the tensor functions.

The gradient in the particle σ(fixed value) and component k is calculated after the sum over all pairs μ and ν

```
In[24]:= PartialD[{x, δ, η, Γ}][
          U[ruu[red@μ, red@ν]], xuu[red@σ, k]]
         GRADIENT = % /. PartialDDistanceRule
```

Out[24]= $\frac{\partial r^{\mu\nu}}{\partial x^{\sigma k}} U'[r^{\mu\nu}]$

Out[25]= $\frac{(x^{\mu k} - x^{\nu k})\,(\delta^{\mu}{}_{\sigma} - \delta^{\nu}{}_{\sigma})\,U'[r^{\mu\nu}]}{r^{\mu\nu}}$

The fully expanded gradient on the first particle $\sigma \to 1$ after a sum over all the interactions is

```
In[26]:= Map[Simplify,
          Table[
              1/2 Sum[GRADIENT, {μ, 1, 3}, {ν, 1, 3}] /. {σ → 1}
              , {k, 1, 3}] /. ruu[i_, j_] :→ ruu[Sequence @@ Sort@{i, j}] // Expand //
           Collect[#, Power[_, _]] &
          ,
          {2}]
```

Out[26]= $\left\{\frac{(x^{11} - x^{21})\,U'[r^{12}]}{r^{12}} + \frac{(x^{11} - x^{31})\,U'[r^{13}]}{r^{13}},\ \frac{(x^{12} - x^{22})\,U'[r^{12}]}{r^{12}} + \frac{(x^{12} - x^{32})\,U'[r^{13}]}{r^{13}},\ \frac{(x^{13} - x^{23})\,U'[r^{12}]}{r^{12}} + \frac{(x^{13} - x^{33})\,U'[r^{13}]}{r^{13}}\right\}$

The Laplacian on the particle $\sigma \to 1$ is then

```
In[27]:= PartialD[{x, δ, η, Γ}][
          U[ruu[red@μ, red@ν]], {xuu[red@σ, k], xuu[red@σ, k]}]
         LAPLACIAN = (% /. PartialDDistanceRule) /. HigherOrderPartialDDistanceRule
```

Out[27]= $\frac{\partial^2 r^{\mu\nu}}{\partial x^{\sigma k}\,\partial x^{\sigma k}} U'[r^{\mu\nu}] + \left(\frac{\partial r^{\mu\nu}}{\partial x^{\sigma k}}\right)^2 U''[r^{\mu\nu}]$

Out[28]= $\left(-\frac{(x^{\mu k} - x^{\nu k})^2\,(\delta^{\mu}{}_{\sigma} - \delta^{\nu}{}_{\sigma})^2}{(r^{\mu\nu})^3} + \frac{(\delta^{\mu}{}_{\sigma} - \delta^{\nu}{}_{\sigma})\,(\delta^{k}{}_{k}\,\delta^{\mu}{}_{\sigma} - \delta^{k}{}_{k}\,\delta^{\nu}{}_{\sigma})}{r^{\mu\nu}}\right) U'[r^{\mu\nu}] + \frac{(x^{\mu k} - x^{\nu k})^2\,(\delta^{\mu}{}_{\sigma} - \delta^{\nu}{}_{\sigma})^2\,U''[r^{\mu\nu}]}{(r^{\mu\nu})^2}$

where we assume an implicit sum over all the interactions and k as well

Test

Expanding the Laplacian in the first particle $\{\sigma \to 1\}$ with inverse square potential

```
In[29]:= (((1/2 Sum[LAPLACIAN /. {σ → 1}, {μ, 1, 3}, {ν, 1, 3}, {k, 1, 3}]) /.
            ruu[i_, j_] ⧴ ruu[Sequence @@ Sort@{i, j}])) /. U → (1/# &)
         Laplacian[1] = Map[FullSimplify, Expand[%] // (Collect[#, Power[_, _]] &)]
```

$$
\begin{aligned}
\text{Out[29]}= \frac{1}{2}\Bigg(&-\frac{(r^{12})^{-1}-\frac{(x^{11}-x^{21})^2}{(r^{12})^3}}{(r^{12})^2}+\frac{2\,(x^{11}-x^{21})^2}{(r^{12})^5}+\frac{2\,(-x^{11}+x^{21})^2}{(r^{12})^5}-\frac{(r^{12})^{-1}-\frac{(-x^{11}+x^{21})^2}{(r^{12})^3}}{(r^{12})^2}-\\
&\frac{(r^{12})^{-1}-\frac{(x^{12}-x^{22})^2}{(r^{12})^3}}{(r^{12})^2}+\frac{2\,(x^{12}-x^{22})^2}{(r^{12})^5}+\frac{2\,(-x^{12}+x^{22})^2}{(r^{12})^5}-\frac{(r^{12})^{-1}-\frac{(-x^{12}+x^{22})^2}{(r^{12})^3}}{(r^{12})^2}-\\
&\frac{(r^{12})^{-1}-\frac{(x^{13}-x^{23})^2}{(r^{12})^3}}{(r^{12})^2}+\frac{2\,(x^{13}-x^{23})^2}{(r^{12})^5}+\frac{2\,(-x^{13}+x^{23})^2}{(r^{12})^5}-\frac{(r^{12})^{-1}-\frac{(-x^{13}+x^{23})^2}{(r^{12})^3}}{(r^{12})^2}-\\
&\frac{(r^{13})^{-1}-\frac{(x^{11}-x^{31})^2}{(r^{13})^3}}{(r^{13})^2}+\frac{2\,(x^{11}-x^{31})^2}{(r^{13})^5}+\frac{2\,(-x^{11}+x^{31})^2}{(r^{13})^5}-\frac{(r^{13})^{-1}-\frac{(-x^{11}+x^{31})^2}{(r^{13})^3}}{(r^{13})^2}-\\
&\frac{(r^{13})^{-1}-\frac{(x^{12}-x^{32})^2}{(r^{13})^3}}{(r^{13})^2}+\frac{2\,(x^{12}-x^{32})^2}{(r^{13})^5}+\frac{2\,(-x^{12}+x^{32})^2}{(r^{13})^5}-\frac{(r^{13})^{-1}-\frac{(-x^{12}+x^{32})^2}{(r^{13})^3}}{(r^{13})^2}-\\
&\frac{(r^{13})^{-1}-\frac{(x^{13}-x^{33})^2}{(r^{13})^3}}{(r^{13})^2}+\frac{2\,(x^{13}-x^{33})^2}{(r^{13})^5}+\frac{2\,(-x^{13}+x^{33})^2}{(r^{13})^5}-\frac{(r^{13})^{-1}-\frac{(-x^{13}+x^{33})^2}{(r^{13})^3}}{(r^{13})^2}\Bigg)
\end{aligned}
$$

$$
\begin{aligned}
\text{Out[30]}= &-\frac{3}{(r^{12})^3}-\frac{3}{(r^{13})^3}+\frac{3\left((x^{11}-x^{21})^2+(x^{12}-x^{22})^2+(x^{13}-x^{23})^2\right)}{(r^{12})^5}+\\
&\frac{3\left((x^{11}-x^{31})^2+(x^{12}-x^{32})^2+(x^{13}-x^{33})^2\right)}{(r^{13})^5}
\end{aligned}
$$

Now using brute force to test the result. The pairwise potential is

```
In[31]:= V[i_, i_] = 0;
         V[i_, j_] = 1/Sqrt[ (x[i] - x[j])^2 + (y[i] - y[j])^2 + (z[i] - z[j])^2 ];
```

And the total potential

```
In[33]:= VTot = FullSimplify /@ Simplify@Sum[V[i, j], {i, 1, 3}, {j, 1, 3}] / 2
```

$$
\begin{aligned}
\text{Out[33]}= &\frac{1}{\sqrt{(x[1]-x[2])^2+(y[1]-y[2])^2+(z[1]-z[2])^2}}+\\
&\frac{1}{\sqrt{(x[1]-x[3])^2+(y[1]-y[3])^2+(z[1]-z[3])^2}}+\\
&\frac{1}{\sqrt{(x[2]-x[3])^2+(y[2]-y[3])^2+(z[2]-z[3])^2}}
\end{aligned}
$$

The Laplacian after simplifications is

```
In[34]:= (D[VTot, x[1], x[1]] + D[VTot, y[1], y[1]] + D[VTot, z[1], z[1]])

     % /. {Power[(x[i_] - x[j_])^2 + __, (-3/2)] → ruu[red@i, red@j]^(-3),
        Power[(x[i_] - x[j_])^2 + __, (-5/2)] → ruu[red@i, red@j]^(-5),
        x[p_] → xuu[red@p, 1], y[p_] → xuu[red@p, 2], z[p_] → xuu[red@p, 3]} //
       (Collect[#, Power[_, _]] &) // Simplify
```

Out[34]=

$$\frac{3\,(x[1]-x[2])^2}{((x[1]-x[2])^2+(y[1]-y[2])^2+(z[1]-z[2])^2)^{5/2}}+\frac{3\,(y[1]-y[2])^2}{((x[1]-x[2])^2+(y[1]-y[2])^2+(z[1]-z[2])^2)^{5/2}}-\frac{3}{((x[1]-x[2])^2+(y[1]-y[2])^2+(z[1]-z[2])^2)^{3/2}}+\frac{3\,(z[1]-z[2])^2}{((x[1]-x[2])^2+(y[1]-y[2])^2+(z[1]-z[2])^2)^{5/2}}+\frac{3\,(x[1]-x[3])^2}{((x[1]-x[3])^2+(y[1]-y[3])^2+(z[1]-z[3])^2)^{5/2}}+\frac{3\,(y[1]-y[3])^2}{((x[1]-x[3])^2+(y[1]-y[3])^2+(z[1]-z[3])^2)^{5/2}}-\frac{3}{((x[1]-x[3])^2+(y[1]-y[3])^2+(z[1]-z[3])^2)^{3/2}}+\frac{3\,(z[1]-z[3])^2}{((x[1]-x[3])^2+(y[1]-y[3])^2+(z[1]-z[3])^2)^{5/2}}$$

Out[35]=

$$-\frac{3}{(r^{12})^3}-\frac{3}{(r^{13})^3}+\frac{3\left((x^{11}-x^{21})^2+(x^{12}-x^{22})^2+(x^{13}-x^{23})^2\right)}{(r^{12})^5}+\frac{3\left((x^{11}-x^{31})^2+(x^{12}-x^{32})^2+(x^{13}-x^{33})^2\right)}{(r^{13})^5}$$

which is the same of the expanded tensor expression we found expanding the tensor-like expression

```
In[36]:= Laplacian[1] // Simplify
```

Out[36]=

$$-\frac{3}{(r^{12})^3}-\frac{3}{(r^{13})^3}+\frac{3\left((x^{11}-x^{21})^2+(x^{12}-x^{22})^2+(x^{13}-x^{23})^2\right)}{(r^{12})^5}+\frac{3\left((x^{11}-x^{31})^2+(x^{12}-x^{32})^2+(x^{13}-x^{33})^2\right)}{(r^{13})^5}$$