założenia

Zakładamy, że A jest macierzą symetryczną o rozmiarze n × n (A∈Matrices[{n , n} , Reals , Symmetric[{1 , 2}]]) oraz, że r1, p1,rk,pk,rkm,pkm są wektorami o rozmiarze n i rzeczywistych współrzędnych ((r1|p1)∈ Vectors[n , Reals]).

```
In[1]:= $Assumptions = (A ∈ Matrices[{n , n} , Reals , Symmetric[{1 , 2}]]) &&
         ((r1 | p1 | rk | pk | rkm | pkm) ∈ Vectors[n , Reals]);
```

pierwszy krok dowodu, sprawdzenie pojedynczego przypadku

$r1^T r1 \overset{\text{można rozpisać}}{\longleftrightarrow} \sum_i r1_i \, r1^i \leftrightarrow$

zwężenie współrzędnych 1 oraz 2 tensora $r1 \otimes r1$,
którego wartość dla współrzędnych k,
l dana jest przez $r1_k \, r1^l$ (TensorContract [r1⊗r1 , {{1 , 2}}])

$A\, p1 \overset{\text{można rozpisać}}{\longleftrightarrow} \sum_k A^i_{\;k} \, p1^k \leftrightarrow$

zwężenie współrzędnych 2 oraz 3 tensora $A \otimes p1$,
którego wartość dla współrzędych l, m ,
n dana jest przez $A^l_{\;m} \, p1^n$ (TensorContract [A⊗p1 , {{2 , 3}}])

$p1^T A\, p1 \overset{\text{można rozpisać}}{\longleftrightarrow} \sum_{i,k} p1_i \, A^i_{\;k} \, p1^k \leftrightarrow$

zwężenie współrzędnych 1, 2 oraz 3, 4 tensora $p1 \otimes A \otimes p1$,
którego wartość dla współrzędnych l, m , n ,
o dana przez $p1_l \, A^m_{\;n} \, p1^o$ (TensorContract [p1⊗A⊗p1 , {{1 , 2} , {3 , 4}}])

```
In[2]:= α1 = TensorContract[r1⊗r1 , {{1 , 2}}] / TensorContract[p1⊗A⊗p1 , {{1 , 2} , {3 , 4}}];
```

```
In[3]:= r2 = r1 - α1 TensorContract[A⊗p1 , {{2 , 3}}];
```

```
In[4]:= β1 = TensorContract[r2⊗r2 , {{1 , 2}}] / TensorContract[r1⊗r1 , {{1 , 2}}];
```

```
In[5]:= p2 = r2 + β1 p1;
```

Zakładamy dodatkowo, że p1 = r1 i podmieniamy wszędzie r1 za p1 aby uprościć wyrażenia (/.{p1 -> r1}).

```
In[6]:= (*twierdzenie pomocnicze*)
```

```
In[7]:= TensorContract[p1⊗A⊗p1 , {{1 , 2}, {3 , 4}}] ==
         TensorContract[p1⊗A⊗r1 , {{1 , 2}, {3 , 4}}] /. {p1 → r1} // TensorReduce // FullSimplify
```

```
Out[7]= True
```

In[8]:= `(*2a*)`

In[9]:= `TensorContract [r2⊗r1 , {{1 , 2}}] /. {p1 → r1} // TensorReduce`

Out[9]= `0`

In[10]:= `(*2b*)`

In[11]:= `TensorContract [r2⊗p1 , {{1 , 2}}] /. {p1 → r1} // TensorReduce`

Out[11]= `0`

In[12]:= `(*2c*)`

In[13]:= `TensorContract [p2⊗A⊗p1 , {{1 , 2}, {3 , 4}}] /. {p1 → r1} // TensorReduce`

Out[13]= `0`

---

indukcja

In[14]:= `αk = TensorContract [rk⊗rk , {{1 , 2}}] / TensorContract [pk⊗A⊗pk , {{1 , 2}, {3 , 4}}];`

In[15]:= `rkp = rk − αk TensorContract [A⊗pk , {{2 , 3}}];`

In[16]:= `βk = TensorContract [rkp⊗rkp , {{1 , 2}}] / TensorContract [rk⊗rk , {{1 , 2}}];`

In[17]:= `pkp = rkp + βk pk;`

In[18]:= `αkm = TensorContract [rkm⊗rkm , {{1 , 2}}] / TensorContract [pkm⊗A⊗pkm , {{1 , 2}, {3 , 4}}];`

In[19]:= `subrk = {rk → rkm − αkm TensorContract [A⊗pkm , {{2 , 3}}]};`

In[20]:= `βkm = TensorContract [rk⊗rk , {{1 , 2}}] / TensorContract [rkm⊗rkm , {{1 , 2}}];`

In[21]:= `subpk = {pk → rk + βkm pkm};`

In[22]:= `(*twierdzenie  pomocnicze*)`

In[23]:= `(*z założenia indukcyjnego TensorContract [A⊗pk⊗pk,{{1,3},{2,4}}]==`
`  TensorContract [A⊗pk⊗rk,{{1,3},{2,4}}]*)`

In[24]:= `TensorContract [pkp⊗A⊗pkp , {{1 , 2}, {3 , 4}}] ==`
`    TensorContract [pkp⊗A⊗rkp , {{1 , 2}, {3 , 4}}] // TensorReduce // FullSimplify`

Out[24]= `((TensorContract [A⊗pk⊗pk, {{1, 3}, {2, 4}}] − TensorContract [A⊗pk⊗rk, {{1, 3}, {2, 4}}])`
`    (TensorContract [A⊗pk⊗pk, {{1, 3}, {2, 4}}]² − 2 TensorContract [A⊗pk⊗pk, {{1, 3}, {2, 4}}] ×`
`        TensorContract [A⊗pk⊗rk, {{1, 3}, {2, 4}}] + TensorContract [rk⊗rk, {{1, 2}}] ×`
`        TensorContract [A⊗A⊗pk⊗pk, {{1, 3}, {2, 5}, {4, 6}}])) /`
`    TensorContract [A⊗pk⊗pk, {{1, 3}, {2, 4}}] == 0`

In[25]:= `(*2a*)`

In[26]:= `(*zgadza  się  jeżeli  skorzystamy  z  pomocniczego  twierdzenia*)`

In[27]:= **TensorContract[rkp⊗rk , {{1 , 2}}] // TensorReduce**

Out[27]= TensorContract[rk⊗rk, {{1, 2}}] −
  (TensorContract[rk⊗rk, {{1, 2}}] × TensorContract[A⊗pk⊗rk, {{1, 3}, {2, 4}}]) /
    TensorContract[A⊗pk⊗pk, {{1, 3}, {2, 4}}]

In[28]:= **(∗2b∗)**

In[29]:= **(∗zgadza się jeżeli skorzystamy z założenia indukcyjnego ∗)**

In[30]:= **TensorContract[rkp⊗pk , {{1 , 2}}] // TensorReduce**

Out[30]= TensorContract[pk⊗rk, {{1, 2}}] − TensorContract[rk⊗rk, {{1, 2}}]

In[31]:= **TensorContract[rkp⊗pk , {{1 , 2}}] /. subpk // TensorReduce // FullSimplify**

Out[31]= (TensorContract[pkm⊗rk, {{1, 2}}] × TensorContract[rk⊗rk, {{1, 2}}]) /
    TensorContract[rkm⊗rkm, {{1, 2}}]

In[32]:= **(∗2c∗)**

In[33]:= **(∗zgadza się jeżeli skorzystamy z pomocniczego twierdzenia ∗)**

In[34]:= **TensorContract[pkp⊗A⊗pk , {{1 , 2}, {3 , 4}}] // TensorReduce**

Out[34]= TensorContract[A⊗pk⊗pk, {{1, 3}, {2, 4}}] − TensorContract[A⊗pk⊗rk, {{1, 3}, {2, 4}}]