```
> restart:
```

We load our package:

```
> with(IntegrableConnections);
```

[*Eigenring*, *Evalm*, *G_matrix*, *G_matrix0*, *HyperexponentialSolutions*, *Id*, *Mapply*, *MatrixColumnPrimpart*, *MatrixColumnPrimpartRat*, *Mexpsolde*, *Mpolsolde*, *Mratsolde*, *Msylvester*, *Mylinsolve*, *NNIexponents*, *PolynomialSolutions*, *RationalSolutions*, *Reduction*, *TestIntegrabilityConditions*, *Trigonalize*, *backsubs*, *candidates*, *col_pval*, *colechelon*, *colval*, *complement*, *convert_back*, *denomatrix*, *denomvector*, *direct_ratsol*, *eval_block_reduced*, *evalam*, *exp_parts*, *formal_reduce*, *gen_sylvester*, *getargs*, *good_form*, *intDiff*, *int_diff_split*, *l_colechelon*, *l_column_rank*, *l_moser_reduce*, *l_qtcd*, *l_super*, *l_super_reduce*, *lc_evala*, *lc_evalam*, *localize*, *log_collect*, *log_free*, *mat_block_reduce*, *mat_change_exp_part*, *mat_check*, *mat_coeff*, *mat_colvaluation*, *mat_convert*, *mat_copy*, *mat_difference*, *mat_eval*, *mat_get_dim*, *mat_get_exp_part*, *mat_get_ext*, *mat_get_indicial_polynomial*, *mat_get_inv_transformation*, *mat_get_order*, *mat_get_ramification*, *mat_get_terms*, *mat_get_transformation*, *mat_get_type*, *mat_lc*, *mat_lead_mon*, *mat_product*, *mat_pval*, *mat_subs*, *mat_sum*, *mat_swap*, *mat_to_simple*, *mat_tools*, *mat_transform*, *mat_val*, *mat_valuation*, *matrix_series*, *matval*, *minimalIntegerRoot*, *mult*, *my_linsolve*, *mydegree*, *myequaind*, *new_matrix*, *newton*, *newton_polygon*, *nonzero*, *normalm*, *pencil_block_reduce*, *pencil_solve*, *pgaussUnimod*, *poly_pval*, *poly_sols*, *poly_val*, *prank*, *pval*, *qtcd*, *ramified_case*, *ramified_reduction*, *randomMatrix*, *randomPoly*, *randomRat*, *ratsuper*, *reduceColRank4*, *reg_sols*, *regular_series*, *remove*, *rowechelon*, *seq_evala*, *sim_sylvester*, *simple_eval*, *simple_sols*, *sortColumns*, *super_form*, *system_expsolde*, *tensor_prod*, *transform*, *truncate*, *utils*, *val*, *val1*, *val_ldegree*, *valuation*, *vectdegree*] **(1)**

The library linalg is also needed.

```
> with(linalg):
```

We load OreModules in order to use the procedure to write a D-finite partial differential system as an integrable connection.

```
> with(OreModules):
```

BrycLetac system in dimension 2

We define the OreAlgebra (needed for OreModules):

```
> Alg2:=DefineOreAlgebra(diff=[d1,x1],diff=[d2,x2],polynom=[x1,x2],
  comm=[d]):
```

We give the equations of the system:

```
> R2 := matrix(2,1,[-(1/2)*d*d2+d1^2-x2*d2^2,2*d1*d2+x1*d2^2]);
```

$$R2 := \begin{bmatrix} -\frac{1}{2}\, d\, d2 + d1^2 - x2\, d2^2 \\ 2\, d1\, d2 + x1\, d2^2 \end{bmatrix} \quad \textbf{(2)}$$

We write the system as an integrable connection:

```
> C2:=OreModules[Connection](R2,Alg2);
```

$$C2 := \left[\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{1}{2}\,x1 \\ 0 & \frac{1}{2}\,d & 0 & x2 \\ 0 & 0 & 0 & \frac{(-3-d)\,x1}{-4\,x2+x1^2} \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{2}\,x1 \\ 0 & 0 & 0 & \frac{6+2\,d}{-4\,x2+x1^2} \end{bmatrix}\right] \quad (3)$$

If we want to know the basis of the associated module in which the connection is written, we can use the procedure KBasis of OreModules:

```
> OreModules[KBasis](R2,Alg2);
```

$$\left[\lambda_1, \lambda_1\,d2, d1\,\lambda_1, \lambda_1\,d2^2\right] \quad (4)$$

We use our procedure for computing hyperexponential solutions of the integrable connection:

```
> HyperexpSols:=HyperexponentialSolutions(C2,[x1,x2],['param',[d]]);
```

$$HyperexpSols := \begin{bmatrix} \frac{1}{4}\,x1^2\,d+x2 & x1 & 1 & \left(-4\,x2+x1^2\right)^{\frac{1}{2}-\frac{1}{2}d} \\ 1 & 0 & 0 & 2\,(d-1)\left(-4\,x2+x1^2\right)^{-\frac{1}{2}d-\frac{1}{2}} \\ \frac{1}{2}\,x1\,d & 1 & 0 & -(d-1)\left(-4\,x2+x1^2\right)^{-\frac{1}{2}d-\frac{1}{2}}\,x1 \\ 0 & 0 & 0 & 4\left(-4\,x2+x1^2\right)^{-\frac{3}{2}-\frac{1}{2}d}\left(-1+d^2\right) \end{bmatrix} \quad (5)$$

The system admits thus three rational solutions and on extra hyperexponential solution given by (x1^2-4*x2)^(1/2-1/2*d).

BrycLetac system in dimension 3

We define the OreAlgebra (needed for OreModules):

```
> Alg3:=DefineOreAlgebra(diff=[d1,x1],diff=[d2,x2],diff=[d3,x3],
  polynom=[x1,x2,x3],comm=[d]):
```

We give the equations of the system:

```
> R3 := matrix(3,1,[-d*d2+d1^2-x2*d2^2-2*x3*d2*d3,-(1/2)*d*d3+2*d1*
  d2+x1*d2^2-x3*d3^2,2*d1*d3+d2^2+2*x1*d2*d3+x2*d3^2]);
```

$$R3 := \begin{bmatrix} -d\,d2+d1^2-x2\,d2^2-2\,x3\,d2\,d3 \\ -\frac{1}{2}\,d\,d3+2\,d2\,d1+x1\,d2^2-x3\,d3^2 \\ 2\,d1\,d3+d2^2+2\,x1\,d2\,d3+x2\,d3^2 \end{bmatrix} \quad (6)$$

We write the system as an integrable connection:

```
> C3:=OreModules[Connection](R3,Alg3):
```

If we want to know the basis of the associated module in which the connection is written, we can use the procedure KBasis of OreModules:

```
> OreModules[KBasis](R3,Alg3);
```

$$\left[\lambda_1, \lambda_1\, d3, \lambda_1\, d2, \lambda_1\, d1, \lambda_1\, d3^2, \lambda_1\, d2\, d3, \lambda_1\, d1\, d3, d3^3\, \lambda_1\right] \tag{7}$$

We use our procedure for computing hyperexponential solutions of the integrable connection:

```
> HyperexpSols:=HyperexponentialSolutions(C3,[x1,x2,x3],['param',[d]]
  );
```

$$HyperexpSols := \begin{bmatrix} \frac{1}{24} d^2\, x1^3 + \frac{1}{4} x2\, x1\, d + x3 & \frac{1}{2} x1^2\, d + x2 & x1 & 1 \\ 1 & 0 & 0 & 0 \\ \frac{1}{4} x1\, d & 1 & 0 & 0 \\ \frac{1}{8} x1^2\, d^2 + \frac{1}{4} d\, x2 & x1\, d & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

Therefore the original system admits four linearly independent rational solutions but no more hyperexponential solutions.

The computation took some time (around 190 seconds). If we want to do this in a more efficient way, then an idea is to use the rational solutions to reduce
the size of the connection before searching for non-rational extra hyperexponential solutions. This can be done as below:

```
> R:=RationalSolutions(C3,[x1,x2,x3],['param',[d]]);
```

$$R := \begin{bmatrix} \frac{1}{6} d^2\, x1^3 + x2\, x1\, d + 4\, x3 & \frac{1}{2} x1^2\, d + x2 & x1 & 1 \\ 4 & 0 & 0 & 0 \\ x1\, d & 1 & 0 & 0 \\ \frac{1}{2} x1^2\, d^2 + d\, x2 & x1\, d & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

we complete the rational solutions into an invertible matrix:

```
> P:=augment(R,stackmatrix(diag(0$4),diag(1$4)));
  invP:=inverse(P);
```

$$P := \begin{bmatrix} \frac{1}{6} d^2 x1^3 + x2\, x1\, d + 4\, x3 & \frac{1}{2} x1^2 d + x2 & x1 & 1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x1\, d & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} x1^2 d^2 + d\, x2 & x1\, d & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$invP := \begin{bmatrix} 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{4} x1\, d & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{8} x1^2 d^2 - \frac{1}{4} d\, x2 & -x1\, d & 1 & 0 & 0 & 0 & 0 \\ 1 & -\frac{1}{24} d^2 x1^3 - x3 + \frac{1}{4} x2\, x1\, d & \frac{1}{2} x1^2 d - x2 & -x1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(10)}$$

We perform the change of variable and look at the structure of the resulting matrices:

```
> for i from 1 to 3 do B||i:= map( factor@normal, evalm( invP&*(C3[i]
  &*P-map(diff,P,x||i)) )) od:
> zerotest:=proc(X) if X=0 then 0 else if X=1 then 1 else x fi; fi;
  end:
> seq(map(zerotest,B||i),i=1..3);
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 & x & x & x & 0 \\ 0 & 0 & 0 & 0 & x & x & x & 0 \\ 0 & 0 & 0 & 0 & x & x & x & 0 \\ 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & x & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & x & 0 \\ 0 & 0 & 0 & 0 & x & x & x & 0 \\ 0 & 0 & 0 & 0 & x & x & x & 0 \\ 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & 1 & 0 \\ 0 & 0 & 0 & 0 & x & x & x & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x \end{bmatrix} \quad \textbf{(11)}$$

Then, we only have to search for hyperexponential solutions of the following 4*4 blocks:

```
> C11:=submatrix(B||1,5..8,5..8):
  C21:=submatrix(B||2,5..8,5..8):
  C31:=submatrix(B||3,5..8,5..8):
> HyperexponentialSolutions([C11,C21,C31],[x1,x2,x3]);
                                    { }                                  (12)
```

Our algorithm outputs that there are no extra hyperexponential solutions so that we find the same result that by running directly the procedure HyperexponentialSolutions but in a faster way!