

```

adPower[x_, k_, Env_] [c_*y_] /; MemberQ[$Basis, y] := CF[c adPower[x, k, Env][y]];
adPower[x_, k_, Env_] [y_Plus] := adPower[x, k, Env] /@ y;
adPower[_ , 0, Env_] [y_] := y;
adPower[x_, k_, Env_] [0] = 0;
adPower[x_, k_, Env_] [y_] /; MemberQ[$Basis, y] :=
  adPower[x, k, Env][y] = B[adPower[x, k - 1, Env][y], x];
adPower[x_, k_] [y_] := adPower[x, k, {$Algebra, $k}][y];

adExp[x_] [y_] /; SSQ[x] := Expand@Module[{A, b, c},
  A = Table[Coefficient[B[b, x], c], {c, $Basis}, {b, $Basis}];
  CF[$Basis.Normal[Series[MatrixExp[A], {ϵ, 0, $k}]] . Table[Coefficient[y, b], {b, $Basis}]]
]

```

```

adExp[x_] [y_] /; NilQ[x] := Expand@Module[{k = 0, s = 0},
  While[adPower[x, k][y] != 0,
    s += adPower[x, k][y] / k!;
    ++k
  ];
  s
]

```

```

λTangent[] = 0;
λTangent[xs___, x_] := adExp[x][λTangent[xs]] + ∂λ x;
λTangent[xs_List] := λTangent @@ xs

```

```

SBasis = SortBy[$Basis, If[#[[1]] == x, {0, #[[3]] - #[[2]]}, {1, #[[2]] - #[[3]]}] &];
sol[-1] = Table[If[SSQ[bb], ((bb /. x → ξ1) + (bb /. x → ξ2)), 0], {bb, $Basis}];
Do[
  Block[{$k = k},
    s0 = Thread[$Basis → sol[k - 1]];
    lhs = λTangent[
      Join[Table[If[NilQ[bb], λ, 1] bb (bb /. {x → ξ1, y → η1}), {bb, $Basis}],
        Table[If[NilQ[bb], λ, 1] bb (bb /. {x → ξ2, y → η2}), {bb, $Basis}]]
    ];
    rhs = λTangent[
      $Basis ($Basis /. {xαβ → (xαβ /. s0) + ek fαβ[λ], yαβ → (yαβ /. s0) + ek gαβ[λ]})
    ];
    eqns = Transpose@{
      (Coefficient[CF[lhs - rhs], #] == 0) & /@ SBasis,
      SBasis /. {xαβ → fαβ[0] == 0, yαβ → gαβ[0] == 0}
    };
    unknowns = SBasis /. {xαβ → fαβ[λ], yαβ → gαβ[λ]};
    s = TriangularDSolve[eqns, unknowns, λ];
    sol[k] = CF[sol[k - 1] + ek ($Basis /. {xαβ → fαβ[λ], yαβ → gαβ[λ]} /. s)]
  ],
  {k, 0, $k}
];
cm[i_, j_ → k_] :=
  Δ2E{i,j}→{k}[sol[$k].$Basis /. {xαβ → xαβ[k], yαβ → yαβ[k], ξ1αβ → ξαβ[i], η1αβ → ηαβ[i],
    ξ2αβ → ξαβ[j], η2αβ → ηαβ[j], λ → 1}];

```