Complexity Estimates for Three Uncoupling Algorithms

Alin Bostan, Frédéric Chyzak, Élie de Panafieu

INRIA, INRIA, LIAFA, DDMF Team

February 21, 2013

How to compute the characteristic polynomial of a matrix?

simple case: companion matrix $C = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix}$

$$\chi(X) = X^n - c_0 - c_1 X - \ldots - c_{n-1} X^{n-1}.$$

How to compute the characteristic polynomial of a matrix?

simple case: companion matrix $C = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix}$

$$\chi(X) = X^n - c_0 - c_1 X - \ldots - c_{n-1} X^{n-1}.$$

reformulation: find P invertible such that PMP^{-1} is companion.

How to compute the characteristic polynomial of a matrix?

simple case: companion matrix $C = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix}$

$$\chi(X) = X^n - c_0 - c_1 X - \ldots - c_{n-1} X^{n-1}.$$

reformulation: find P invertible such that PMP^{-1} is companion.

[Krylov31]: [Keller-Gehrig85]'s algorithm, for a random row vector u

$$P := \begin{bmatrix} u \\ uM \\ \vdots \\ uM^{n-1} \end{bmatrix}; \qquad PM = \begin{bmatrix} uM \\ uM^2 \\ \vdots \\ uM^n \end{bmatrix} = CP.$$

How to compute the characteristic polynomial of a matrix?

simple case: companion matrix $C = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix}$

$$\chi(X) = X^n - c_0 - c_1 X - \ldots - c_{n-1} X^{n-1}.$$

reformulation: find P invertible such that PMP^{-1} is companion.

[Krylov31]: if $uM^k \in \mathrm{Vect}(u, uM, \ldots, uM^{k-1})$, then

$P := \begin{bmatrix} u \\ \vdots \\ uM^{k-1} \\ e_\star \\ \vdots \\ e_\star \end{bmatrix}$; $PMP^{-1} = \left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right]$; factor of degree k of $\chi(X)$.

How to compute the characteristic polynomial of a matrix?

simple case: companion matrix $C = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix}$

$$\chi(X) = X^n - c_0 - c_1 X - \ldots - c_{n-1} X^{n-1}.$$

reformulation: find P invertible such that PMP^{-1} is companion.

[Danilevski37]: pivot operations like in the Gaussian elimination

$\left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right]$; $\begin{bmatrix} C_1 & & \\ & \ddots & \\ & & C_t \end{bmatrix}$; partial factorisation of $\chi(X)$.

How to uncouple a differential system $\partial Y = MY$?

simple case: companion matrix $\partial Y = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix} Y$

$$\partial^n y_1 = c_0 y_1 + c_1 \partial y_1 + \ldots + c_{n-1} \partial^{n-1} y_1.$$

How to uncouple a differential system $\partial Y = MY$?

simple case: companion matrix $\partial Y = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix} Y$

$$\partial^n y_1 = c_0 y_1 + c_1 \partial y_1 + \ldots + c_{n-1} \partial^{n-1} y_1.$$

reformulation: find P invertible such that $Z := PY$ satisfies $\partial Z = CZ$.

How to uncouple a differential system $\partial Y = MY$?

simple case: companion matrix $\partial Y = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix} Y$

$$\partial^n y_1 = c_0 y_1 + c_1 \partial y_1 + \ldots + c_{n-1} \partial^{n-1} y_1.$$

reformulation: find P invertible such that $Z := PY$ statisfies $\partial Z = CZ$.

Cyclic Vector Method: [Schlesinger08], [Cope36]

for a random row vector u

$$P := \begin{bmatrix} u \\ \Delta u \\ \vdots \\ \Delta^{n-1} u \end{bmatrix}; \qquad \Delta P = CP.$$

How to uncouple a differential system $\partial Y = MY$?

simple case: companion matrix $\partial Y = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix} Y$

$$\partial^n y_1 = c_0 y_1 + c_1 \partial y_1 + \ldots + c_{n-1} \partial^{n-1} y_1.$$

reformulation: find P invertible such that $Z := PY$ statisfies $\partial Z = CZ$.

Cyclic Vector Method: if $\Delta^k u \in \mathrm{Vect}(u, \Delta u, \ldots, \Delta^{k-1} u)$, then

$P := \begin{bmatrix} u \\ \vdots \\ \Delta^{k-1} u \\ e_\star \\ \vdots \\ e_\star \end{bmatrix}$; $\Delta(P)P^{-1} = \left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right]$; differential equation of order k.

How to uncouple a differential system $\partial Y = MY$?

simple case: companion matrix $\partial Y = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix} Y$

$$\partial^n y_1 = c_0 y_1 + c_1 \partial y_1 + \ldots + c_{n-1} \partial^{n-1} y_1 .$$

reformulation: find P invertible such that $Z := PY$ statisfies $\partial Z = CZ$.

[Deligne70], [Katz87], [Adjamagbo88], [Churchill-Kovacic02]:
compute u such that $\{u, \Delta u, \ldots, \Delta^{n-1} u\}$ is free.

How to uncouple a differential system $\partial Y = MY$?

simple case: companion matrix $\partial Y = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix} Y$

$$\partial^n y_1 = c_0 y_1 + c_1 \partial y_1 + \ldots + c_{n-1} \partial^{n-1} y_1 .$$

reformulation: find P invertible such that $Z := PY$ satisfies $\partial Z = CZ$.

[Barkatou93] and [Zürcher94]: pivot operations like in the Gaussian elimination

$\left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right]$; $\begin{bmatrix} C_1 & & \\ & \ddots & \\ & & C_t \end{bmatrix}$; several differential equations.

Motivations

- Apply algorithms that input differential equations to differential systems,
- preliminary work for the comparison with direct algorithms computing the rational solutions of a differential system,
- understand the links between the existing uncoupling algorithms,
- canonical shape for matrices in pseudo-linear Ore-algebras.

Contributions

Analysis of three uncoupling algorithms: CVM, DBZ and AZ

- new algebraic analysis of DBZ and AZ for general inputs,
- precise complexity analysis of DBZ and AZ for generic inputs $\tilde{\mathcal{O}}(n^5 \deg(M))$,
- fast algorithm for CVM $\tilde{\mathcal{O}}(n^{\omega+1} \deg(M))$,
- magma implementation and benchmarks.

Uncoupling and companion matrices

Uncoupling

Transformation of a differential system

$$\partial Y = MY$$

where M is a matrix in $\mathrm{Mat}_{n,n}(\mathbb{K}_d[x])$ and Y a vector of unknowns,

Uncoupling

Transformation of a differential system

$$\partial Y = MY$$

where M is a matrix in $\mathrm{Mat}_{n,n}(\mathbb{K}_d[x])$ and Y a vector of unknowns,

into one or several differential equations

$$\partial^k y = c_0 y + c_1 \partial y + \cdots + c_{k-1} \partial^{k-1} y$$

where the c_i are rational functions,

Uncoupling

Transformation of a differential system

$$\partial Y = MY$$

where M is a matrix in $\mathrm{Mat}_{n,n}(\mathbb{K}_d[x])$ and Y a vector of unknowns,

into one or several differential equations

$$\partial^k y = c_0 y + c_1 \partial y + \cdots + c_{k-1} \partial^{k-1} y$$

where the c_i are rational functions,

and a correspondance between the solutions of the system and of the equation.

The companion case

When M is a companion matrix, the transformation is simple: the system $\partial Y = MY$ becomes

$$\partial \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

The companion case

When M is a companion matrix, the transformation is simple: the system $\partial Y = MY$ becomes

$$\partial \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

which implies

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \partial y_1 \\ \vdots \\ \partial^{n-1} y_1 \end{bmatrix}$$

The companion case

When M is a companion matrix, the transformation is simple: the system $\partial Y = MY$ becomes

$$\partial \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

which implies

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \partial y_1 \\ \vdots \\ \partial^{n-1} y_1 \end{bmatrix}$$

and

$$\partial^n y_1 = c_0 y_1 + c_1 \partial y_1 + \cdots + c_{n-1} \partial^{n-1} y_1 .$$

The companion case

When M is a companion matrix, the transformation is simple: the system $\partial Y = MY$ becomes

$$\partial \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

which implies

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \partial y_1 \\ \vdots \\ \partial^{n-1} y_1 \end{bmatrix}$$

and

$$\partial^n y_1 = c_0 y_1 + c_1 \partial y_1 + \cdots + c_{n-1} \partial^{n-1} y_1 .$$

Bijection between the solutions of the system and of the equation.

Block decomposition

$\partial Y = CY \qquad \Longleftrightarrow$ differential equation of order n

Block decomposition

$$\partial Y = CY \qquad \Longleftrightarrow \text{ differential equation of order } n$$

$$\partial Y = \begin{array}{c} \begin{array}{cc} \leftarrow k \rightarrow & \end{array} \\ \begin{bmatrix} C & 0 \\ \star & \star \end{bmatrix} \end{array} Y \qquad \Longleftrightarrow \text{ differential equation of order } k$$

Block decomposition

$$\partial Y = CY \qquad \Longleftrightarrow \text{ differential equation of order } n$$

$$\begin{array}{cc} & \leftarrow k \rightarrow \quad \\ \partial Y = & \begin{bmatrix} C & 0 \\ \star & \star \end{bmatrix} Y \end{array} \qquad \Longleftrightarrow \text{ differential equation of order } k$$

$$\begin{array}{cc} & \leftarrow k_1 \rightarrow \qquad\qquad \leftarrow k_t \rightarrow \\ \partial Y = & \begin{bmatrix} C_1 & & \\ & \ddots & \\ & & C_t \end{bmatrix} Y \end{array}$$

$\Longleftrightarrow$ t differential equations of orders $k_1, \ldots, k_t$

Differential change of basis

In the differential system $\partial Y = MY$

the linear change of variable $Z = PY$

produces the new differential system $\partial Z = \left(PMP^{-1} + (\partial P)P^{-1}\right) Z.$

Differential change of basis

In the differential system $\partial Y = MY$

the linear change of variable $Z = PY$

produces the new differential system $\partial Z = \left(PMP^{-1} + (\partial P)P^{-1}\right) Z.$

Differential change of variable of M by P: $P[M] := PMP^{-1} + (\partial P)P^{-1}.$

Differential change of basis

In the differential system	$\partial Y = MY$
the linear change of variable	$Z = PY$
produces the new differential system	$\partial Z = \left(PMP^{-1} + (\partial P)P^{-1}\right) Z.$
Differential change of variable of M by P:	$P[M] := PMP^{-1} + (\partial P)P^{-1}.$

Reformulation of the uncoupling problem:
find P such that $P[M] = \left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right]$.

The Cyclic Vector Method

The Cyclic Vector Method (CVM)

Choose a row vector u and set $\Delta = v \mapsto vM + \partial v$.

Find k maximal such that F is free $F = \{u,\ \Delta u,\ \ldots,\ \Delta^{k-1} u\}$

and complement it into a basis $F \cup \{a_{k+1}, \ldots, a_n\}$.

The Cyclic Vector Method (CVM)

Choose a row vector u and set $\Delta = v \mapsto vM + \partial v.$

Find k maximal such that F is free $F = \{u,\ \Delta u,\ \ldots,\ \Delta^{k-1} u\}$

and complement it into a basis $F \cup \{a_{k+1}, \ldots, a_n\}.$

The matrix $P_{\mathrm{CVM}} = \begin{bmatrix} u \\ \vdots \\ \Delta^{k-1} u \\ a_{k+1} \\ \vdots \\ a_n \end{bmatrix}$ then satisfies $\Delta P_{\mathrm{CVM}} = \begin{bmatrix} \Delta u \\ \vdots \\ \Delta^{k} u \\ \star \\ \vdots \\ \star \end{bmatrix},$

The Cyclic Vector Method (CVM)

Choose a row vector u and set $\Delta = v \mapsto vM + \partial v.$

Find k maximal such that F is free $F = \{u,\ \Delta u,\ \ldots,\ \Delta^{k-1} u\}$

and complement it into a basis $F \cup \{a_{k+1}, \ldots, a_n\}.$

The matrix $P_{\text{CVM}} = \begin{bmatrix} u \\ \vdots \\ \Delta^{k-1} u \\ a_{k+1} \\ \vdots \\ a_n \end{bmatrix}$ then satisfies $\Delta P_{\text{CVM}} = \left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right] P_{\text{CVM}},$

The Cyclic Vector Method (CVM)

Choose a row vector u and set $\Delta = v \mapsto vM + \partial v$.

Find k maximal such that F is free $F = \{u,\ \Delta u,\ \ldots,\ \Delta^{k-1} u\}$

and complement it into a basis $F \cup \{a_{k+1}, \ldots, a_n\}$.

The matrix $P_{\mathrm{CVM}} = \begin{bmatrix} u \\ \vdots \\ \Delta^{k-1} u \\ a_{k+1} \\ \vdots \\ a_n \end{bmatrix}$ then satisfies $P_{\mathrm{CVM}} M + P'_{\mathrm{CVM}} = \left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right] P_{\mathrm{CVM}}$,

The Cyclic Vector Method (CVM)

Choose a row vector u and set $\Delta = v \mapsto vM + \partial v$.

Find k maximal such that F is free $F = \{u,\ \Delta u,\ \ldots,\ \Delta^{k-1} u\}$

and complement it into a basis $F \cup \{a_{k+1}, \ldots, a_n\}$.

The matrix $P_{\mathrm{CVM}} = \begin{bmatrix} u \\ \vdots \\ \Delta^{k-1} u \\ a_{k+1} \\ \vdots \\ a_n \end{bmatrix}$ then satisfies $(P_{\mathrm{CVM}} M + P'_{\mathrm{CVM}}) {P_{\mathrm{CVM}}}^{-1} = \left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right],$

The Cyclic Vector Method (CVM)

Choose a row vector u and set $\Delta = v \mapsto vM + \partial v$.

Find k maximal such that F is free $F = \{u,\ \Delta u,\ \ldots,\ \Delta^{k-1} u\}$

and complement it into a basis $F \cup \{a_{k+1}, \ldots, a_n\}$.

The matrix $P_{\text{CVM}} = \begin{bmatrix} u \\ \vdots \\ \Delta^{k-1} u \\ a_{k+1} \\ \vdots \\ a_n \end{bmatrix}$ then satisfies $P_{\text{CVM}}[M] = \left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right]$.

The Cyclic Vector Method (CVM)

Choose a row vector u and set $\Delta = v \mapsto vM + \partial v$.

Find k maximal such that F is free $F = \{u,\ \Delta u,\ \ldots,\ \Delta^{k-1} u\}$

and complement it into a basis $F \cup \{a_{k+1}, \ldots, a_n\}$.

The matrix $P_{\mathrm{CVM}} = \begin{bmatrix} u \\ \vdots \\ \Delta^{k-1} u \\ a_{k+1} \\ \vdots \\ a_n \end{bmatrix}$ then satisfies $P_{\mathrm{CVM}}[M] = \left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right]$.

Furthermore, one can compute u such that $P_{\mathrm{CVM}}[M] = C$.
(survey in [Churchill-Kovacic02])

Remarks and experimental observations of CVM

CVM has bad reputation: its output is said to be “very complicated” in comparison to other uncoupling methods [Hilali83], [Barkatou93], [Zürcher94], [Abramov99], [Gerhold02].

Remarks and experimental observations of CVM

CVM has bad reputation: its output is said to be "very complicated" in comparison to other uncoupling methods [Hilali83], [Barkatou93], [Zürcher94], [Abramov99], [Gerhold02].

Experimental observations for random intputs M and u:

1. $P_{\mathrm{CVM}}[M] = C$ instead of $\begin{bmatrix} C & 0 \\ \star & \star \end{bmatrix}$,
2. $\deg(P_{\mathrm{CVM}}) = (n-1)\deg(M) + \deg(u)$,
3. $\deg(C) = \frac{n(n+1)}{2}\deg(M) + n\deg(u)$.

Remarks and experimental observations of CVM

CVM has bad reputation: its output is said to be “very complicated” in comparison to other uncoupling methods [Hilali83], [Barkatou93], [Zürcher94], [Abramov99], [Gerhold02].

Experimental observations for random intputs M and u:

1 $P_{\mathrm{CVM}}[M] = C$ instead of $\left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right]$,

2 $\deg(P_{\mathrm{CVM}}) = (n-1)\deg(M) + \deg(u)$,

3 $\deg(C) = \frac{n(n+1)}{2}\deg(M) + n\deg(u)$.

Explanations:

1 for a generic M, the family $\{u, \Delta u, \ldots, \Delta^{n-1} u\}$ is free,

2,3 for any row vector v, with equality in the generic case

$$\deg(\Delta v) = \deg(vM + \partial v) \leq \deg(v) + \deg(M)$$

[Cluzeau03]

The Danilevski-Barkatou-Zürcher algorithm

Step I:

$$M \to \begin{bmatrix} C & 0 \\ \star & \star \end{bmatrix}$$

Step II:

$$\begin{bmatrix} C & 0 \\ \star & \star \end{bmatrix} \to \begin{bmatrix} C & 0 \\ 0 & \star \end{bmatrix} \to \begin{bmatrix} C_1 & & \\ & \ddots & \\ & & C_t \end{bmatrix}$$

or starts over and treats at least one more row.

The pivot operation

Differential change of basis by elementary matrices

$$E_{i,j}(a) = \begin{array}{cc} & \begin{array}{cccc} & j & & \\ & \downarrow & & \end{array} \\ \begin{array}{c} \\ \\ i \rightarrow \\ \\ \end{array} & \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & a & \ddots & \\ & & & 1 \end{bmatrix} \end{array} \qquad E_{i,j}(a)[M] = \begin{array}{cc} & \begin{array}{ccccc} & j & \leftarrow & i & \end{array} \\ \begin{array}{c} \\ j \\ \downarrow \\ i \\ \\ \end{array} & \begin{bmatrix} & | & & & \\ & | & & & \\ & | & & & \\ - & \star & - & - & - \\ & | & & & \end{bmatrix} \end{array}$$

The pivot operation

Differential change of basis by elementary matrices

$$E_{i,j}(a) = \begin{array}{cc} & \begin{matrix} & \overset{j}{\downarrow} & & \end{matrix} \\ \begin{matrix} \\ \\ i \rightarrow \\ \\ \end{matrix} & \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & a & \ddots & \\ & & & 1 \end{bmatrix} \end{array} \qquad E_{i,j}(a)[M] = \begin{array}{cc} & \begin{matrix} & j & \leftarrow & i & \end{matrix} \\ \begin{matrix} \\ j \\ \downarrow \\ i \\ \\ \end{matrix} & \begin{bmatrix} & | & & & \\ & | & & & \\ & | & & & \\ - & \star & - & - & - \\ & | & & & \end{bmatrix} \end{array}$$

$$E_{i,i}(a) = \begin{array}{cc} & \begin{matrix} & \overset{i}{\downarrow} & & \end{matrix} \\ \begin{matrix} \\ i \rightarrow \\ \\ \\ \end{matrix} & \begin{bmatrix} 1 & & & \\ & a & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \end{array} \qquad E_{i,i}(a)[M] = \begin{array}{cc} & \begin{matrix} & i & & \end{matrix} \\ \begin{matrix} \\ i \\ \\ \\ \end{matrix} & \begin{bmatrix} & | & & \\ - & \star & - & - \\ & | & & \\ & | & & \end{bmatrix} \end{array}$$

The Danilevski-Barkatou-Zürcher algorithm (DBZ): Step I

$$\begin{bmatrix} \star & \star & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix} \quad \begin{bmatrix} \star & 1 & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \star & \star & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \star & \star & 1 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & \star & 1 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}$$

The Danilevski-Barkatou-Zürcher algorithm (DBZ): Step I

$$
\begin{bmatrix} \star & \star & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} \star & 1 & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
$$

$$
\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \star & \star & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \star & \star & 1 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & \star & 1 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
$$

When the upper-diagonal coefficient is 0, invert it with a non-zero coefficient further on the same row.

The Danilevski-Barkatou-Zürcher algorithm (DBZ): Step I

$$
\begin{bmatrix} \star & \star & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} \star & 1 & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
$$

$$
\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \star & \star & \star & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \star & \star & 1 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & \star & 1 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & \star \\ \star & \star & \star & \cdots & \star \\ \vdots & \vdots & \vdots & & \vdots \\ \star & \star & \star & \cdots & \star \end{bmatrix}
$$

When the upper-diagonal coefficient is 0, invert it with a non-zero coefficient further on the same row. If there are none, we have reached

$$
\left[\begin{array}{cccc|ccc}
0 & 1 & & & 0 & \cdots & 0 \\
 & & \ddots & & \vdots & & \vdots \\
0 & & 0 & 1 & 0 & \cdots & 0 \\
\star & \cdots & \cdots & \star & & & \\
\hline
\star & \cdots & \cdots & \star & \star & \cdots & \star \\
\vdots & & & \vdots & \vdots & & \vdots \\
\star & \cdots & \cdots & \star & \star & \cdots & \star
\end{array}\right]
= \begin{bmatrix} C & 0 \\ \star & \star \end{bmatrix}.
$$

DBZ: Step II

By elementary differential changes of basis, M then reaches the shape:

$$\begin{bmatrix} \begin{matrix} & & \\ & C & \\ & & \end{matrix} & \begin{matrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{matrix} \\ \begin{matrix} \star & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \star & 0 & \cdots & 0 \end{matrix} & \star \end{bmatrix}.$$

DBZ: Step II

By elementary differential changes of basis, M then reaches the shape:

$$\left[\begin{array}{cccc|ccc} & & & & 0 & \cdots & 0 \\ & C & & & \vdots & & \vdots \\ & & & & 0 & \cdots & 0 \\ \star & 0 & \cdots & 0 & & & \\ \vdots & \vdots & & \vdots & & \star & \\ \star & 0 & \cdots & 0 & & & \end{array}\right].$$

If all the $\star$ are 0s, then the matrix is $\left[\begin{smallmatrix} C & 0 \\ 0 & \star \end{smallmatrix}\right]$ and DBZ is applied to the lower-right block, leading eventually to the decomposition $\begin{bmatrix} C_1 & & \\ & \ddots & \\ & & C_t \end{bmatrix}$.

DBZ: Step II

By elementary differential changes of basis, M then reaches the shape:

$$\begin{bmatrix} & C & & & 0 & \cdots & 0 \\ & & & & \vdots & & \vdots \\ & & & & 0 & \cdots & 0 \\ \star & 0 & \cdots & 0 & & & \\ \vdots & \vdots & & \vdots & & \star & \\ \star & 0 & \cdots & 0 & & & \end{bmatrix}.$$

If all the $\star$ are 0s, then the matrix is $\left[\begin{smallmatrix} C & 0 \\ 0 & \star \end{smallmatrix}\right]$ and DBZ is applied to the lower-right block, leading eventually to the decomposition $\begin{bmatrix} C_1 & & \\ & \ddots & \\ & & C_t \end{bmatrix}$.

Else a cyclic change of basis is applied

$$\begin{bmatrix} \star & 1 & 0 & \cdots & 0 & \star & \cdots & \star \\ 0 & & & & & 0 & \cdots & 0 \\ \vdots & & C & & & \vdots & & \vdots \\ 0 & & & & & 0 & \cdots & 0 \\ \star & 0 & \cdots & \cdots & 0 & \star & \cdots & \star \\ \vdots & \vdots & & & \vdots & \vdots & & \vdots \\ \star & 0 & \cdots & \cdots & 0 & \star & \cdots & \star \end{bmatrix}$$

DBZ: Step II

By elementary differential changes of basis, M then reaches the shape:

$$\left[\begin{array}{cccc|ccc} & & C & & 0 & \cdots & 0 \\ & & & & \vdots & & \vdots \\ & & & & 0 & \cdots & 0 \\ \star & 0 & \cdots & 0 & & & \\ \vdots & \vdots & & \vdots & & \star & \\ \star & 0 & \cdots & 0 & & & \end{array}\right].$$

If all the $\star$ are 0s, then the matrix is $\left[\begin{smallmatrix} C & 0 \\ 0 & \star \end{smallmatrix}\right]$ and DBZ is applied to the lower-right block, leading eventually to the decomposition $\begin{bmatrix} C_1 & & \\ & \ddots & \\ & & C_t \end{bmatrix}$.

Else a cyclic change of basis is applied

$$\begin{bmatrix} \star & 1 & 0 & \cdots & 0 & \star & \cdots & \star \\ 0 & & & & & 0 & \cdots & 0 \\ \vdots & & C & & & \vdots & & \vdots \\ 0 & & & & & 0 & \cdots & 0 \\ \star & 0 & \cdots & \cdots & 0 & \star & \cdots & \star \\ \vdots & \vdots & & & \vdots & \vdots & & \vdots \\ \star & 0 & \cdots & \cdots & 0 & \star & \cdots & \star \end{bmatrix}$$

DBZ starts over. The particular shape of this matrix ensures that one more row will be treated, which proves that the algorithm terminates.

Experimental observations of DBZ

Tests of DBZ for random inputs.

Expectations	**Observations**
diagonal block-companion	one companion matrix
exponential growth of the degrees (as observed in the Gaussian elimination algorithm)	quadratic growth of the degrees differential Bareiss phenomenon ?

Experimental observations of DBZ

Tests of DBZ for random inputs.

Expectations	**Observations**
diagonal block-companion	one companion matrix
exponential growth of the degrees (as observed in the Gaussian elimination algorithm)	quadratic growth of the degrees differential Bareiss phenomenon ?

$$M^{(1)} = E[M] \qquad \deg(M^{(1)}) \leq 3d \qquad \deg(M^{(1)}) = 3d$$

$$M^{(2)} = E^{(1)}[M^{(1)}] \qquad \deg(M^{(2)}) \leq 9d \qquad \deg(M^{(2)}) = 6d$$

$$M^{(3)} = E^{(2)}[M^{(2)}] \qquad \deg(M^{(3)}) \leq 27d \qquad \deg(M^{(3)}) = 10d$$

$$M^{(4)} = E^{(3)}[M^{(3)}] \qquad \deg(M^{(4)}) \leq 81d \qquad \deg(M^{(4)}) = 15d$$

The Abramov-Zima algorithm

Step I:

$$\partial Y = MY \to \partial Z = \left[\begin{smallmatrix} T & 0 \\ \star & \star \end{smallmatrix}\right] Z$$

Step II:

1. extract a differential equation that cancels $e_1 Y$,
2. solve this equation,
3. inject the solutions into the initial differential system,
4. start over.

The Abramov-Zima algorithm (AZ): Step I

$$\partial \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \star & \star & \cdots & \star \\ \star & \cdots & \cdots & \star \\ \vdots & & & \vdots \\ \star & \cdots & \cdots & \star \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad z_2 = \begin{bmatrix} \star & \cdots & \star \end{bmatrix} \begin{bmatrix} y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$z_1 := y_1$

$$\partial \begin{bmatrix} z_1 \\ z_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \star & 1 & & & \\ \star & \star & \star & \cdots & \star \\ \star & \cdots & \cdots & \cdots & \star \\ \vdots & & & & \vdots \\ \star & \cdots & \cdots & \cdots & \star \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} \qquad z_3 = \begin{bmatrix} \star & \cdots & \star \end{bmatrix} \begin{bmatrix} y_3 \\ \vdots \\ y_n \end{bmatrix}$$

$$\partial \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \star & 1 & & & \\ \star & \star & 1 & & \\ \star & \cdots & \cdots & \cdots & \star \\ \vdots & & & & \vdots \\ \star & \cdots & \cdots & \cdots & \star \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix}$$

When the upper-diagonal coefficient is 0, invert it with a non-zero coefficient further on the same row. If there are none, the matrix has shape $\begin{bmatrix} T & 0 \\ \star & \star \end{bmatrix}$,

where $T = \begin{bmatrix} \star & 1 & & \\ \vdots & \ddots & \ddots & \\ \vdots & & \ddots & 1 \\ \star & \cdots & \cdots & \star \end{bmatrix}$.

The Abramov-Zima algorithm (AZ): Step I

$$\partial \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \star & \star & \cdots & \star \\ \star & \cdots & \cdots & \star \\ \vdots & & & \vdots \\ \star & \cdots & \cdots & \star \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$\begin{bmatrix} z_1 \\ z_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & & & \\ \star & \cdots & \cdots & \star \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$z_1 := y_1$

$$\partial \begin{bmatrix} z_1 \\ z_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \star & 1 & & & \\ \star & \star & \star & \cdots & \star \\ \star & \cdots & \cdots & \cdots & \star \\ \vdots & & & & \vdots \\ \star & \cdots & \cdots & \cdots & \star \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & \star & \cdots & \cdots & \star \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

$$\partial \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \star & 1 & & & \\ \star & \star & 1 & & \\ \star & \cdots & \cdots & \cdots & \star \\ \vdots & & & & \vdots \\ \star & \cdots & \cdots & \cdots & \star \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix}$$

When the upper-diagonal coefficient is 0, invert it with a non-zero coefficient further on the same row. If there are none, the matrix has shape $\begin{bmatrix} T & 0 \\ \star & \star \end{bmatrix}$,

where $T = \begin{bmatrix} \star & 1 & & \\ \vdots & \ddots & \ddots & \\ \vdots & & \ddots & 1 \\ \star & \cdots & \cdots & \star \end{bmatrix}$.

AZ: Step II

1. Extract from

$$\partial \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{bmatrix} = \begin{bmatrix} \star & 1 & & \\ \vdots & \ddots & \ddots & \\ \vdots & & \ddots & 1 \\ \star & \cdots & \cdots & \star \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{bmatrix}$$

   a differential equation of order k that cancels $z_1 = y_1$.
   For example, pivot operations to shape T into a companion matrix.
2. Solve it and inject the solutions into the initial differential system.
3. AZ is then applied to this new smaller system.

AZ: Step II

1. Extract from
$$\partial \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{bmatrix} = \begin{bmatrix} \star & 1 & & \\ \vdots & \ddots & \ddots & \\ \vdots & & \ddots & 1 \\ \star & \cdots & \cdots & \star \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{bmatrix}$$
a differential equation of order k that cancels $z_1 = y_1$.
For example, pivot operations to shape T into a companion matrix.
2. Solve it and inject the solutions into the initial differential system.
3. AZ is then applied to this new smaller system.

Remarks:

Step I the matrix of change of basis is $U \cdot \mathrm{Perm}$ where U is upper-triangular and Perm is a permutation matrix,

Step II the matrix of change of basis L is lower-triangular with 1s on its diagonal.

Experimental observations of AZ

Tests of AZ for random inputs.

Expectations	**Observations**
block-decomposition $\begin{bmatrix} T & 0 \\ \star & \star \end{bmatrix}$	only one block T
exponential growth of the degrees	quadratic growth of the degrees
	same output as DBZ and CVM (e_1)
	the matrices L and U are the LU decomposition of the matrix of change of basis computed by DBZ.

Algebraic Analysis

Algebraic interpretation of CVM

Let u be a row vector, CVM computes the differential equation of smallest order that cancels uY for any solution of $\partial Y = MY$.

$$\partial(uY) = u\partial(Y) + \partial(u)Y = (uM + \partial u)\, Y = \Delta(u)Y$$

Algebraic interpretation of CVM

Let u be a row vector, CVM computes the differential equation of smallest order that cancels uY for any solution of $\partial Y = MY$.

$$\partial(uY) = u\partial(Y) + \partial(u)Y = (uM + \partial u)\, Y = \Delta(u)Y$$

$$\partial^k(uY) = c_0(uY) + c_1\partial(uY) + \cdots + c_{k-1}\partial^{k-1}(uY)$$

Algebraic interpretation of CVM

Let u be a row vector, CVM computes the differential equation of smallest order that cancels uY for any solution of $\partial Y = MY$.

$$\partial(uY) = u\partial(Y) + \partial(u)Y = (uM + \partial u)\, Y = \Delta(u)Y$$

$$\left(\Delta^k u\right) Y = c_0 uY + c_1(\Delta u)Y + \cdots + c_{k-1}(\Delta^{k-1}u)Y$$

Algebraic interpretation of CVM

Let u be a row vector, CVM computes the differential equation of smallest order that cancels uY for any solution of $\partial Y = MY$.

$$\partial(uY) = u\partial(Y) + \partial(u)Y = (uM + \partial u)\, Y = \Delta(u)Y$$

$$\left(\Delta^k u\right) Y = \left(c_0 u + c_1 \Delta u + \cdots + c_{k-1}\Delta^{k-1} u\right) Y$$

by the Cauchy-Lipschitz a.k.a. Picard-Lindelöf Theorem,
there is a fundamental system of solutions.

Algebraic interpretation of CVM

Let u be a row vector, CVM computes the differential equation of smallest order that cancels uY for any solution of $\partial Y = MY$.

$$\partial(uY) = u\partial(Y) + \partial(u)Y = (uM + \partial u)\,Y = \Delta(u)Y$$

$$\Delta^k u = c_0 u + c_1 \Delta u + \cdots + c_{k-1}\Delta^{k-1} u$$

by the Cauchy-Lipschitz a.k.a. Picard-Lindelöf Theorem
there is a fundamental system of solutions.

Algebraic interpretation of CVM

Let u be a row vector, CVM computes the differential equation of smallest order that cancels uY for any solution of $\partial Y = MY$.

$$\partial(uY) = u\partial(Y) + \partial(u)Y = (uM + \partial u)\, Y = \Delta(u)Y$$

$$\Delta^k u = \begin{bmatrix} c_0 & \cdots & c_{k-1} \end{bmatrix} \begin{bmatrix} u \\ \Delta u \\ \vdots \\ \Delta^{k-1} u \end{bmatrix}$$

Algebraic interpretation of CVM

In particular, the differential equation of smallest order that cancels $e_1 Y$ for every solution of $\partial Y = MY$

$$\partial^k y = c_0 y + c_1 \partial y + \cdots + c_{k-1} \partial^{k-1} y$$

is characterized by

- $(e_1, \Delta e_1, \ldots, \Delta^{k-1} e_1)$ is free,
- $\Delta^k e_1 = \begin{bmatrix} c_0 & \cdots & c_{k-1} \end{bmatrix} \begin{bmatrix} e_1 \\ \Delta e_1 \\ \vdots \\ \Delta^{k-1} e_1 \end{bmatrix}.$

Structure of the output of DBZ and AZ

Theorem: DBZ$^{(\mathrm{I})}$, AZ and CVM (e_1) compute the differential equation of smallest order that cancels $e_1 Y$ for every solution of $\partial Y = MY$.

Structure of the output of DBZ and AZ

Theorem: DBZ$^{(I)}$, AZ and CVM (e_1) compute the differential equation of smallest order that cancels $e_1 Y$ for every solution of $\partial Y = MY$.

Proof:

- DBZ$^{(I)}$ and AZ both compute a matrix of change of basis P such that, if $\partial Y = MY$ and $Z = PY$, then

$$\partial Z = \left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right] Z$$

  where C is a companion matrix of dimension denoted by k,

Structure of the output of DBZ and AZ

Theorem: DBZ$^{(\mathrm{I})}$, AZ and CVM (e_1) compute the differential equation of smallest order that cancels $e_1 Y$ for every solution of $\partial Y = MY$.

Proof:

- DBZ$^{(\mathrm{I})}$ and AZ both compute a matrix of change of basis P such that, if $\partial Y = MY$ and $Z = PY$, then

$$\partial Z = \left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right] Z$$

  where C is a companion matrix of dimension denoted by k,

- because of the shape of this system, the differential equation encoded by C cancels $e_1 Z$ and has minimal order,

Structure of the output of DBZ and AZ

Theorem: DBZ$^{(\mathrm{I})}$, AZ and CVM (e_1) compute the differential equation of smallest order that cancels $e_1 Y$ for every solution of $\partial Y = MY$.

Proof:

- DBZ$^{(\mathrm{I})}$ and AZ both compute a matrix of change of basis P such that, if $\partial Y = MY$ and $Z = PY$, then

$$\partial Z = \left[\begin{smallmatrix} C & 0 \\ \star & \star \end{smallmatrix}\right] Z$$

  where C is a companion matrix of dimension denoted by k,

- because of the shape of this system, the differential equation encoded by C cancels $e_1 Z$ and has minimal order,

- in both cases, the first row of P is e_1, because all the matrices of change of basis involved have first row e_1, so

$$e_1 Z = e_1 Y.$$

Intermediate matrices of DBZ and AZ

More precisely, when treating the ith row, the change of basis computed by $\mathrm{DBZ}^{(\mathrm{I})}$ is of the form

$$\begin{bmatrix} e_1 \\ \Delta e_1 \\ \vdots \\ \Delta^{i-1} e_1 \\ e_\star \\ \vdots \\ e_\star \end{bmatrix} \times \text{Permutation Matrix}\,.$$

Intermediate matrices of DBZ and AZ

More precisely, when treating the ith row, the change of basis computed by $\mathrm{DBZ}^{(\mathrm{I})}$ is of the form

$$\begin{bmatrix} e_1 \\ \Delta e_1 \\ \vdots \\ \Delta^{i-1} e_1 \\ e_\star \\ \vdots \\ e_\star \end{bmatrix} \times \text{Permutation Matrix}\,.$$

Similarly, the matrices $P_{\mathrm{AZ}}^{(\mathrm{I})}$ and $P_{\mathrm{AZ}}^{(\mathrm{II})}$ computed by AZ when treating the ith row at Step I or II are the LU decomposition of some

$$\begin{bmatrix} e_1 \\ \vdots \\ \Delta^{i-1} e_1 \\ e_\star \\ \vdots \\ e_\star \end{bmatrix}.$$

Complexity analysis for a generic input

Degree and complexity analysis of DBZ for a generic input

Let $\partial Z = P^{(i)}[M]Z$ denote the differential system manipulated by DBZ after the ith row has been treated, then

$$\deg(P^{(i)}) = \mathcal{O}(i \deg(M)),$$
$$\deg(P^{(i)}[M]) = \mathcal{O}(i^2 \deg(M))$$

by matrix inversion of P.

Degree and complexity analysis of DBZ for a generic input

Let $\partial Z = P^{(i)}[M]Z$ denote the differential system manipulated by DBZ after the ith row has been treated, then

$$\deg(P^{(i)}) = \mathcal{O}(i \deg(M)),$$
$$\deg(P^{(i)}[M]) = \mathcal{O}(i^2 \deg(M))$$

by matrix inversion of P.

Therefore, the modification of the ith row has complexity

$$\tilde{\mathcal{O}}(n^2 i^2 \deg(M))$$

Degree and complexity analysis of DBZ for a generic input

Let $\partial Z = P^{(i)}[M]Z$ denote the differential system manipulated by DBZ after the ith row has been treated, then

$$\deg(P^{(i)}) = \mathcal{O}(i \deg(M)),$$
$$\deg(P^{(i)}[M]) = \mathcal{O}(i^2 \deg(M))$$

by matrix inversion of P.
Therefore, the modification of the ith row has complexity

$$\tilde{\mathcal{O}}(n^2 i^2 \deg(M))$$

and, by summation, for a generic input

$$\text{Complexity DBZ} = \tilde{\mathcal{O}}(n^5 \deg(M)).$$

Degree and complexity analysis of AZ

For a generic input M, the same analysis as for DBZ leads to

$$\text{Complexity AZ} = \tilde{\mathcal{O}}(n^5 \deg(M))$$

using the lemma from [Bareiss68] that the LU decomposition of a matrix of dimension n and degree d satisfies

$$\deg(L), \deg(U), \deg(L^{-1}), \deg(U^{-1}) = \mathcal{O}(nd).$$

Complexity analysis of CVM for a generic input

$P := [\, u \,]$

$v := \Delta u$

while v is not in LeftImage(P), do

$P := \left[\begin{smallmatrix} P \\ v \end{smallmatrix} \right]$

$v := vM + \partial v$

$C := \Delta(P)P^{-1}$

return P and c

Complexity analysis of CVM for a generic input

$P := [\, u \,]$

$v := \Delta u$

while v is not in LeftImage(P), do

$\quad P := \left[\begin{smallmatrix} P \\ v \end{smallmatrix} \right]$

$\quad v := vM + \partial v$

$C := \Delta(P)P^{-1}$

return P and c

$\Delta v = vM + \partial v$

$\deg(\Delta v) \leq \deg(v) + \deg(M)$

Assume $\deg(u) = 0$.

$\deg(v) \leq n \deg(M)$

Complexity analysis of CVM for a generic input

$P := [\,u\,]$

$v := \Delta u$

while v is not in LeftImage(P), do

$\quad P := \left[\begin{smallmatrix} P \\ v \end{smallmatrix}\right]$

$\quad v := vM + \partial v$

$C := \Delta(P)P^{-1}$

return P and c

$\Delta v = vM + \partial v$

$\deg(\Delta v) \leq \deg(v) + \deg(M)$

Assume $\deg(u) = 0$.

$\deg(v) \leq n \deg(M)$

$\deg(P) \leq n \deg(M)$

$\deg(c) \leq n^2 \deg(M)$

$\mathrm{size}(P) = \mathcal{O}(n^3 \deg(M))$

$\mathrm{size}(c) = \mathcal{O}(n^3 \deg(M))$

Complexity analysis of CVM for a generic input

$P := [\,u\,]$

$v := \Delta u$

while v is not in LeftImage(P), do

$P := \left[\begin{smallmatrix} P \\ v \end{smallmatrix}\right]$

$v := vM + \partial v$

$C := \Delta(P)P^{-1}$

return P and c

$\Delta v = vM + \partial v$

$\deg(\Delta v) \leq \deg(v) + \deg(M)$

Assume $\deg(u) = 0$.

$\deg(v) \leq n\deg(M)$

$\deg(P) \leq n\deg(M)$

$\deg(c) \leq n^2\deg(M)$

$\mathrm{size}(P) = \mathcal{O}(n^3\deg(M))$

$\mathrm{size}(c) = \mathcal{O}(n^3\deg(M))$

Complexity analysis of CVM for a generic input

$P := [\, u \,]$

$v := \Delta u$

while v is not in LeftImage(P), do

$\qquad P := \left[\begin{smallmatrix} P \\ v \end{smallmatrix} \right]$

$\qquad v := vM + \partial v$ balanced vector-matrix product

$C := \Delta(P)P^{-1}$

return P and c

$$[\quad nd \quad] \to \left[\quad d \quad \right]$$

$\Delta v = vM + \partial v$

$\deg(\Delta v) \leq \deg(v) + \deg(M)$

Assume $\deg(u) = 0$.

$\deg(v) \leq n \deg(M)$

$\deg(P) \leq n \deg(M)$

$\deg(c) \leq n^2 \deg(M)$

size$(P) = \mathcal{O}(n^3 \deg(M))$

size$(c) = \mathcal{O}(n^3 \deg(M))$

Complexity analysis of CVM for a generic input

$P := [\,u\,]$

$v := \Delta u$

while v is not in LeftImage(P), do $\qquad \tilde{\mathcal{O}}(n^{\omega+1}\deg(M))$

$\qquad P := \left[\begin{smallmatrix} P \\ v \end{smallmatrix}\right]$

$\qquad v := vM + \partial v \qquad \tilde{\mathcal{O}}(n^{\omega}\deg(M))$

$C := \Delta(P)P^{-1}$

return P and c $\qquad [\quad nd \quad] \to \left[\quad d \quad\right]$

$\Delta v = vM + \partial v$

$\deg(\Delta v) \leq \deg(v) + \deg(M)$

Assume $\deg(u) = 0$.

$\deg(v) \leq n\deg(M)$

$\deg(P) \leq n\deg(M)$

$\deg(c) \leq n^2\deg(M)$

size$(P)= \mathcal{O}(n^3\deg(M))$

size$(c)= \mathcal{O}(n^3\deg(M))$

Complexity analysis of CVM for a generic input

$P := [\, u \,]$

$v := \Delta u$

while v is not in LeftImage(P), do $\qquad \tilde{\mathcal{O}}(n^{\omega+1}\deg(M))$

$\qquad P := \begin{bmatrix} P \\ v \end{bmatrix}$

$\qquad v := vM + \partial v \qquad \tilde{\mathcal{O}}(n^{\omega}\deg(M))$

$C := \Delta(P)P^{-1}$

return P and c

$\Delta v = vM + \partial v$

$\deg(\Delta v) \leq \deg(v) + \deg(M)$

Assume $\deg(u) = 0$.

$\deg(v) \leq n\deg(M)$

$\deg(P) \leq n\deg(M)$

$\deg(c) \leq n^2\deg(M)$

$\mathrm{size}(P) = \mathcal{O}(n^3\deg(M))$

$\mathrm{size}(c) = \mathcal{O}(n^3\deg(M))$

Complexity analysis of CVM for a generic input

$P := [\, u \,]$

$v := \Delta u$

while v is not in LeftImage(P), do $\tilde{\mathcal{O}}(n^{\omega+1}\deg(M))$

$\quad P := \begin{bmatrix} P \\ v \end{bmatrix}$

$\quad v := vM + \partial v$ $\tilde{\mathcal{O}}(n^{\omega}\deg(M))$

$c :=$ row vector solution of $v = cP$

return P and c

$\Delta v = vM + \partial v$

$\deg(\Delta v) \leq \deg(v) + \deg(M)$

Assume $\deg(u) = 0$.

$\deg(v) \leq n\deg(M)$

$\deg(P) \leq n\deg(M)$

$\deg(c) \leq n^2\deg(M)$

$\mathrm{size}(P) = \mathcal{O}(n^3\deg(M))$

$\mathrm{size}(c) = \mathcal{O}(n^3\deg(M))$

Complexity analysis of CVM for a generic input

$P := [\,u\,]$

$v := \Delta u$

while v is not in LeftImage(P), do $\tilde{\mathcal{O}}(n^{\omega+1}\deg(M))$

$P := \left[\begin{smallmatrix} P \\ v \end{smallmatrix}\right]$

$v := vM + \partial v$ $\tilde{\mathcal{O}}(n^{\omega}\deg(M))$

$c :=$ row vector solution of $v = cP$ [Storjohann02]'s algorithm

return P and c

$\Delta v = vM + \partial v$

$\deg(\Delta v) \leq \deg(v) + \deg(M)$

Assume $\deg(u) = 0$.

$\deg(v) \leq n\deg(M)$

$\deg(P) \leq n\deg(M)$

$\deg(c) \leq n^2\deg(M)$

size$(P) = \mathcal{O}(n^3\deg(M))$

size$(c) = \mathcal{O}(n^3\deg(M))$

Complexity analysis of CVM for a generic input

$P := [\, u \,]$

$v := \Delta u$

while v is not in LeftImage(P), do $\tilde{\mathcal{O}}(n^{\omega+1}\deg(M))$

$\qquad P := \left[\begin{smallmatrix} P \\ v \end{smallmatrix}\right]$

$\qquad v := vM + \partial v$ $\tilde{\mathcal{O}}(n^{\omega}\deg(M))$

$c :=$ row vector solution of $v = cP$ $\tilde{\mathcal{O}}(n^{\omega+1}\deg(M))$

return P and c

$\Delta v = vM + \partial v$

$\deg(\Delta v) \leq \deg(v) + \deg(M)$

Assume $\deg(u) = 0$.

$\deg(v) \leq n \deg(M)$

$\deg(P) \leq n \deg(M)$

$\deg(c) \leq n^2 \deg(M)$

size$(P) = \mathcal{O}(n^3 \deg(M))$

size$(c) = \mathcal{O}(n^3 \deg(M))$

Implementation and benchmarks

- Magma implementation of DBZ and variants of CVM,

Implementation and benchmarks

- Magma implementation of DBZ and variants of CVM,

- the experimental exponent
  fit the theoretical complexities $c \times \deg(M)^e \times n^p$ for n but not for $\deg(M)$,

					$n = 100$	$n = 5$	$n = 30$
Algorithm	c	e	p		$d = 1$	$d = 100$	$d = 30$
CVM	$6.8\ 10^{-7}$	1.81	$\omega + 1$	3.88	103	3.53	155
DBZ	$7.5\ 10^{-8}$	1.61	5	6.01	∞	2.3	14409
BalConstr	$2.4\ 10^{-6}$	1.01	$\omega + 1$	3.00	12.55	0.5	2.7
NaiveConstr	$3.3\ 10^{-9}$	1.90	4	4.00	1.24	0.2	1.64
StorjohannSolve	$8.2\ 10^{-7}$	1.75	$\omega + 1$	3.87	83.60	3.48	153
NaiveSolve	$4.8\ 10^{-8}$	1.52	5	6.22	106352	0.85	13806

The exponents are obtained by linear regression on suitable domains for each algorithm.

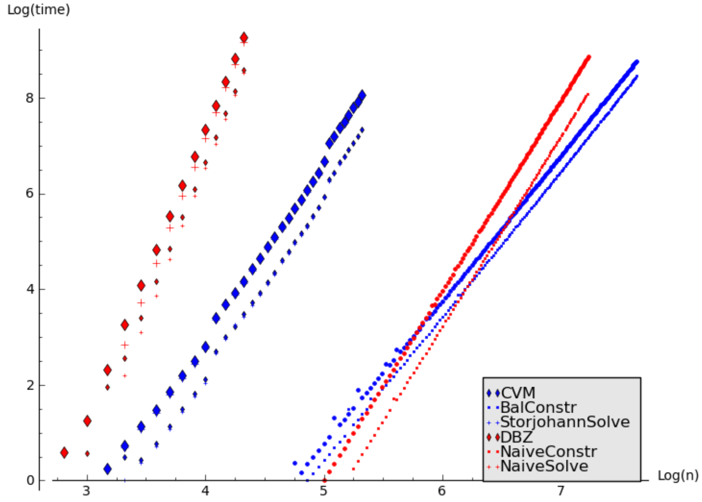
Implementation and benchmarks

- Magma implementation of DBZ and variants of CVM,
- the experimental exponent fit the theoretical complexities $c \times \deg(M)^e \times n^p$ for n but not for $\deg(M)$,
  The exponents are obtained by linear regression on suitable domains for each algorithm.
- the resolution step dominates the timings,

Implementation and benchmarks

- Magma implementation of DBZ and variants of CVM,
- the experimental exponent fit the theoretical complexities $c \times \deg(M)^e \times n^p$ for n but not for $\deg(M)$,
  The exponents are obtained by linear regression on suitable domains for each algorithm.
- the resolution step dominates the timings,
- space consumption limits this implementation.

Implementation and benchmarks



Timings on input matrices of dimension n and coefficients with fixed degree $d = 15$ (smaller marks) or $d = 20$ (larger marks)

Extensions and futur works

- Extension of the degree and complexity analysis to other Ore algebras over rational functions fields, like the finite differences case, and to inhomogeneous differential systems.

Extensions and futur works

- Extension of the degree and complexity analysis to other Ore algebras over rational functions fields, like the finite differences case, and to inhomogeneous differential systems.

- We have achieved analgebraic analysis leading to complexity analysis.

Extensions and futur works

- Extension of the degree and complexity analysis to other Ore algebras over rational functions fields, like the finite differences case, and to inhomogeneous differential systems.
- We have achieved analgebraic analysis leading to complexity analysis.
- Conversely, we are developing a new algorithm combining the general diagonal companion-block decomposition of DBZ with CVM formalism.

Extensions and futur works

- Extension of the degree and complexity analysis to other Ore algebras over rational functions fields, like the finite differences case, and to inhomogeneous differential systems.
- We have achieved analgebraic analysis leading to complexity analysis.
- Conversely, we are developing a new algorithm combining the general diagonal companion-block decomposition of DBZ with CVM formalism.
- non-generic matrices:
  - structured,
  - sparse,
  - find matrices that decompose into several companion blocks.