The MeatAxe

Max Neunhöffer

University of St Andrews

GAC 2010, Allahabad

Introduction

Let $\mathbb{F}$ be a field and $\mathbb{F}^{d \times d}$ the set of $d \times d$-matrices.

Definition ($\mathbb{F}$-algebra, matrix algebra)

An $\mathbb{F}$-algebra is a ring $\mathcal{A}$ with identity together with a ring homomorphism $\iota : \mathbb{F} \to C(\mathcal{A})$ into the centre of $\mathcal{A}$.

Introduction

Let $\mathbb{F}$ be a field and $\mathbb{F}^{d \times d}$ the set of $d \times d$-matrices.

Definition ($\mathbb{F}$-algebra, matrix algebra)

An $\mathbb{F}$-algebra is a ring $\mathcal{A}$ with identity together with a ring homomorphism $\iota : \mathbb{F} \to C(\mathcal{A})$ into the centre of $\mathcal{A}$.
An $\mathbb{F}$-subspace $\mathcal{A}$ of $\mathbb{F}^{d \times d}$ with $1 \in \mathcal{A}$ which is closed under matrix multiplication is called a matrix algebra.

Introduction

Let $\mathbb{F}$ be a field and $\mathbb{F}^{d \times d}$ the set of $d \times d$-matrices.

Definition ($\mathbb{F}$-algebra, matrix algebra)

An $\mathbb{F}$-algebra is a ring $\mathcal{A}$ with identity together with a ring homomorphism $\iota : \mathbb{F} \to C(\mathcal{A})$ into the centre of $\mathcal{A}$.

An $\mathbb{F}$-subspace $\mathcal{A}$ of $\mathbb{F}^{d \times d}$ with $1 \in \mathcal{A}$ which is closed under matrix multiplication is called a matrix algebra.

For a subset $\mathcal{M} \subseteq \mathcal{A}$ we denote by $\langle \mathcal{M} \rangle_{\mathrm{Alg}}$ the intersection of all subalgebras in $\mathcal{A}$ containing $\mathcal{M}$, the algebra generated by $\mathcal{M}$.

Introduction

Let $\mathbb{F}$ be a field and $\mathbb{F}^{d \times d}$ the set of $d \times d$-matrices.

Definition ($\mathbb{F}$-algebra, matrix algebra)

An $\mathbb{F}$-algebra is a ring $\mathcal{A}$ with identity together with a ring homomorphism $\iota : \mathbb{F} \to C(\mathcal{A})$ into the centre of $\mathcal{A}$.
An $\mathbb{F}$-subspace $\mathcal{A}$ of $\mathbb{F}^{d \times d}$ with $1 \in \mathcal{A}$ which is closed under matrix multiplication is called a matrix algebra.
For a subset $\mathcal{M} \subseteq \mathcal{A}$ we denote by $\langle \mathcal{M} \rangle_{\mathrm{Alg}}$ the intersection of all subalgebras in $\mathcal{A}$ containing $\mathcal{M}$, the algebra generated by $\mathcal{M}$.

Definition (Right $\mathcal{A}$-module)

Let $\mathcal{A}$ be an $\mathbb{F}$-algebra. An $\mathbb{F}$-vector space V with a bilinear map $\mu : V \times \mathcal{A} \to V$ is called a right $\mathcal{A}$-module, if

Introduction

Let $\mathbb{F}$ be a field and $\mathbb{F}^{d \times d}$ the set of $d \times d$-matrices.

Definition ($\mathbb{F}$-algebra, matrix algebra)

An $\mathbb{F}$-algebra is a ring $\mathcal{A}$ with identity together with a ring homomorphism $\iota : \mathbb{F} \to C(\mathcal{A})$ into the centre of $\mathcal{A}$.
An $\mathbb{F}$-subspace $\mathcal{A}$ of $\mathbb{F}^{d \times d}$ with $1 \in \mathcal{A}$ which is closed under matrix multiplication is called a matrix algebra.
For a subset $\mathcal{M} \subseteq \mathcal{A}$ we denote by $\langle \mathcal{M} \rangle_{\mathrm{Alg}}$ the intersection of all subalgebras in $\mathcal{A}$ containing $\mathcal{M}$, the algebra generated by $\mathcal{M}$.

Definition (Right $\mathcal{A}$-module)

Let $\mathcal{A}$ be an $\mathbb{F}$-algebra. An $\mathbb{F}$-vector space V with a bilinear map $\mu : V \times \mathcal{A} \to V$ is called a right $\mathcal{A}$-module, if

- $\mu(v, 1_{\mathcal{A}}) = v$ for all $v \in V$ and

Introduction

Let $\mathbb{F}$ be a field and $\mathbb{F}^{d \times d}$ the set of $d \times d$-matrices.

Definition ($\mathbb{F}$-algebra, matrix algebra)

An $\mathbb{F}$-algebra is a ring $\mathcal{A}$ with identity together with a ring homomorphism $\iota : \mathbb{F} \to C(\mathcal{A})$ into the centre of $\mathcal{A}$.
An $\mathbb{F}$-subspace $\mathcal{A}$ of $\mathbb{F}^{d \times d}$ with $1 \in \mathcal{A}$ which is closed under matrix multiplication is called a matrix algebra.
For a subset $\mathcal{M} \subseteq \mathcal{A}$ we denote by $\langle \mathcal{M} \rangle_{\mathrm{Alg}}$ the intersection of all subalgebras in $\mathcal{A}$ containing $\mathcal{M}$, the algebra generated by $\mathcal{M}$.

Definition (Right $\mathcal{A}$-module)

Let $\mathcal{A}$ be an $\mathbb{F}$-algebra. An $\mathbb{F}$-vector space V with a bilinear map $\mu : V \times \mathcal{A} \to V$ is called a right $\mathcal{A}$-module, if

- $\mu(v, 1_{\mathcal{A}}) = v$ for all $v \in V$ and
- $\mu(\mu(v, X), Y) = \mu(v, XY)$ for all $v \in V$ and $X, Y \in \mathcal{A}$.

$\mathcal{A}$-modules

Example (Natural module)

If $\mathcal{A} \leq \mathbb{F}^{d \times d}$ is a matrix algebra, then $V := \mathbb{F}^{1 \times d}$ is a right $\mathcal{A}$-module with $\mu(v, X) := v \cdot X$. It is called the natural module.

$\mathcal{A}$-modules

Example (Natural module)

If $\mathcal{A} \leq \mathbb{F}^{d \times d}$ is a matrix algebra, then $V := \mathbb{F}^{1 \times d}$ is a right $\mathcal{A}$-module with $\mu(v, X) := v \cdot X$. It is called the natural module.

Definition (Submodules and quotient modules)

Let V be an $\mathcal{A}$-module. An $\mathcal{A}$-submodule is an $\mathcal{A}$-invariant subspace $W \leq V$, that is, $W\mathcal{A} = W$.

$\mathcal{A}$-modules

Example (Natural module)

If $\mathcal{A} \leq \mathbb{F}^{d \times d}$ is a matrix algebra, then $V := \mathbb{F}^{1 \times d}$ is a right $\mathcal{A}$-module with $\mu(v, X) := v \cdot X$. It is called the natural module.

Definition (Submodules and quotient modules)

Let V be an $\mathcal{A}$-module. An $\mathcal{A}$-submodule is an $\mathcal{A}$-invariant subspace $W \leq V$, that is, $W\mathcal{A} = W$.
If $W \leq V$ is a submodule, then the quotient space V/W is an $\mathcal{A}$-module with $(v + W)X := vX + W$.

$\mathcal{A}$-modules

Example (Natural module)

If $\mathcal{A} \leq \mathbb{F}^{d \times d}$ is a matrix algebra, then $V := \mathbb{F}^{1 \times d}$ is a right $\mathcal{A}$-module with $\mu(v, X) := v \cdot X$. It is called the natural module.

Definition (Submodules and quotient modules)

Let V be an $\mathcal{A}$-module. An $\mathcal{A}$-submodule is an $\mathcal{A}$-invariant subspace $W \leq V$, that is, $W\mathcal{A} = W$.
If $W \leq V$ is a submodule, then the quotient space V/W is an $\mathcal{A}$-module with $(v + W)X := vX + W$.
A module V is called irreducible if its only submodules are $\{0\}$ and V itself.

$\mathcal{A}$-modules

Example (Natural module)

If $\mathcal{A} \leq \mathbb{F}^{d \times d}$ is a matrix algebra, then $V := \mathbb{F}^{1 \times d}$ is a right $\mathcal{A}$-module with $\mu(v, X) := v \cdot X$. It is called the natural module.

Definition (Submodules and quotient modules)

Let V be an $\mathcal{A}$-module. An $\mathcal{A}$-submodule is an $\mathcal{A}$-invariant subspace $W \leq V$, that is, $W\mathcal{A} = W$.
If $W \leq V$ is a submodule, then the quotient space V/W is an $\mathcal{A}$-module with $(v + W)X := vX + W$.
A module V is called irreducible if its only submodules are $\{0\}$ and V itself.
A composition series for V is a chain of submodules

$$\{0\} = V_{\ell+1} < V_\ell < V_{\ell-1} < \cdots < V_1 = V$$

such that all V_i/V_{i+1} are irreducible.

$\mathcal{A}$-modules on the computer

Let V be an $\mathcal{A}$-module for the $\mathbb{F}$-algebra

$$\mathcal{A} = \langle A_1, \ldots, A_k \rangle_{\mathrm{Alg}} .$$

$\mathcal{A}$-modules on the computer

Let V be an $\mathcal{A}$-module for the $\mathbb{F}$-algebra

$$\mathcal{A} = \langle A_1, \ldots, A_k \rangle_{\mathrm{Alg}} .$$

Then each generator A_i induces a linear map $A_i : V \to V$.

$\mathcal{A}$-modules on the computer

Let V be an $\mathcal{A}$-module for the $\mathbb{F}$-algebra

$$\mathcal{A} = \langle A_1, \ldots, A_k \rangle_{\mathrm{Alg}} .$$

Then each generator A_i induces a linear map $A_i : V \to V$.

Fact

To describe this situation to a computer, it is enough to choose an $\mathbb{F}$-basis $(v_1, \ldots, v_d)$ of V and store one $d \times d$-matrix for each A_i.

GAP examples

see other window

Available methods from Linear Algebra

We can efficiently

- compute in vector spaces and matrix algebras.

Available methods from Linear Algebra

We can efficiently

- compute in vector spaces and matrix algebras.
- in particular multiply vectors with matrices and matrices with matrices.

Available methods from Linear Algebra

We can efficiently

- compute in vector spaces and matrix algebras.
- in particular multiply vectors with matrices and matrices with matrices.
- describe subspaces by bases.

Available methods from Linear Algebra

We can efficiently

- compute in vector spaces and matrix algebras.
- in particular multiply vectors with matrices and matrices with matrices.
- describe subspaces by bases.
- solve systems of linear equations.

Available methods from Linear Algebra

We can efficiently

- compute in vector spaces and matrix algebras.
- in particular multiply vectors with matrices and matrices with matrices.
- describe subspaces by bases.
- solve systems of linear equations.
- compute kernels of matrices.

Available methods from Linear Algebra

We can efficiently

- compute in vector spaces and matrix algebras.
- in particular multiply vectors with matrices and matrices with matrices.
- describe subspaces by bases.
- solve systems of linear equations.
- compute kernels of matrices.
- compute sums and intersections of subspaces given by bases.

Available methods from Linear Algebra

We can efficiently

- compute in vector spaces and matrix algebras.
- in particular multiply vectors with matrices and matrices with matrices.
- describe subspaces by bases.
- solve systems of linear equations.
- compute kernels of matrices.
- compute sums and intersections of subspaces given by bases.
- test membership of a vector in a subspace.

Available methods from Linear Algebra

We can efficiently

- compute in vector spaces and matrix algebras.
- in particular multiply vectors with matrices and matrices with matrices.
- describe subspaces by bases.
- solve systems of linear equations.
- compute kernels of matrices.
- compute sums and intersections of subspaces given by bases.
- test membership of a vector in a subspace.
- transpose matrices.

Available methods from Linear Algebra

We can efficiently

- compute in vector spaces and matrix algebras.
- in particular multiply vectors with matrices and matrices with matrices.
- describe subspaces by bases.
- solve systems of linear equations.
- compute kernels of matrices.
- compute sums and intersections of subspaces given by bases.
- test membership of a vector in a subspace.
- transpose matrices.
- compute characteristic and minimal polynomials.

Available methods from Linear Algebra

We can efficiently

- compute in vector spaces and matrix algebras.
- in particular multiply vectors with matrices and matrices with matrices.
- describe subspaces by bases.
- solve systems of linear equations.
- compute kernels of matrices.
- compute sums and intersections of subspaces given by bases.
- test membership of a vector in a subspace.
- transpose matrices.
- compute characteristic and minimal polynomials.

All these algorithms have time-complexity at most $O(d^3)$ in the dimension d.

Arithmetic over finite fields

For small finite fields we can store a field element using only a few bits!

Arithmetic over finite fields

For small finite fields we can store a field element using only a few bits!
This has several advantages:

Arithmetic over finite fields

For small finite fields we can store a field element using only a few bits!

This has several advantages:

- We save memory.

Arithmetic over finite fields

For small finite fields we can store a field element using only a few bits!

This has several advantages:

- We save memory.
- Since basic field operations are simple, quite often the runtime is dominated by memory accesses.

Arithmetic over finite fields

For small finite fields we can store a field element using only a few bits!

This has several advantages:

- We save memory.
- Since basic field operations are simple, quite often the runtime is dominated by memory accesses. This saves time as well.

Arithmetic over finite fields

For small finite fields we can store a field element using only a few bits!

This has several advantages:

- We save memory.
- Since basic field operations are simple, quite often the runtime is dominated by memory accesses. This saves time as well.
- We can execute several field operations using one processor word operation.

Arithmetic over finite fields

For small finite fields we can store a field element using only a few bits!

This has several advantages:

- We save memory.
- Since basic field operations are simple, quite often the runtime is dominated by memory accesses. This saves time as well.
- We can execute several field operations using one processor word operation.

Example time and memory usage:

Operation	Time		Memory	
	C	U	C	U
Mult. in $\mathbb{F}_2^{4370\times4370}$	320 ms	1335 s	2.3 MB	152 MB
Add. in $\mathbb{F}_2^{1\times4370}$	240 ns	209 μs	550 B	35 kB
Mult. in $\mathbb{F}_3^{500\times500}$	50 ms	2140 ms	78 kB	2 MB

Spinning up

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.

Spinning up

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.

Problem (Module generated by a vector)

Given $0 \neq v \in V$*, find a basis for*

$$\begin{aligned} v\mathcal{A} &:= \{vX \mid X \in \mathcal{A}\} \\ &:= \textit{intersection of all } \mathcal{A}\textit{-submodules containing } v \end{aligned}$$

Spinning up

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.

Problem (Module generated by a vector)

Given $0 \neq v \in V$*, find a basis for*

$$v\mathcal{A} := \{vX \mid X \in \mathcal{A}\} := \textit{intersection of all } \mathcal{A}\textit{-submodules containing } v$$

Solution: the spinning up procedure

1. Initialise $\mathcal{B} := [v]$ and $i := 1$

Spinning up

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.

Problem (Module generated by a vector)

Given $0 \neq v \in V$*, find a basis for*

$$\begin{aligned} v\mathcal{A} &:= \{vX \mid X \in \mathcal{A}\} \\ &:= \textit{intersection of all } \mathcal{A}\textit{-submodules containing } v \end{aligned}$$

Solution: the spinning up procedure

1. Initialise $\mathcal{B} := [v]$ and $i := 1$
2. While $i \leq \text{Length}(\mathcal{B})$ do

Spinning up

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.

Problem (Module generated by a vector)

Given $0 \neq v \in V$*, find a basis for*

$$v\mathcal{A} := \{vX \mid X \in \mathcal{A}\}$$

$$:= \textit{intersection of all } \mathcal{A}\textit{-submodules containing } v$$

Solution: the spinning up procedure

1. Initialise $\mathcal{B} := [v]$ and $i := 1$
2. While $i \leq \text{Length}(\mathcal{B})$ do
3. For j from 1 to k do

Spinning up

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.

Problem (Module generated by a vector)

Given $0 \neq v \in V$, find a basis for

$$\begin{aligned} v\mathcal{A} &:= \{vX \mid X \in \mathcal{A}\} \\ &:= \textit{intersection of all } \mathcal{A}\textit{-submodules containing } v \end{aligned}$$

Solution: the spinning up procedure

1. Initialise $\mathcal{B} := [v]$ and $i := 1$
2. While $i \leq \text{Length}(\mathcal{B})$ do
3. For j from 1 to k do
4. If $y := \mathcal{B}[i] \cdot A_j \notin \langle \mathcal{B} \rangle_{\mathbb{F}}$ then

Spinning up

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.

Problem (Module generated by a vector)

Given $0 \neq v \in V$, *find a basis for*

$$\begin{aligned} v\mathcal{A} &:= \{vX \mid X \in \mathcal{A}\} \\ &:= \text{intersection of all } \mathcal{A}\text{-submodules containing } v \end{aligned}$$

Solution: the spinning up procedure

1. Initialise $\mathcal{B} := [v]$ and $i := 1$
2. While $i \leq \text{Length}(\mathcal{B})$ do
3. For j from 1 to k do
4. If $y := \mathcal{B}[i] \cdot A_j \notin \langle \mathcal{B} \rangle_{\mathbb{F}}$ then
5. Append y to the end of $\mathcal{B}$

Spinning up

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.

Problem (Module generated by a vector)

Given $0 \neq v \in V$, find a basis for

$$v\mathcal{A} := \{vX \mid X \in \mathcal{A}\} := \textit{intersection of all } \mathcal{A}\textit{-submodules containing } v$$

Solution: the spinning up procedure

1. Initialise $\mathcal{B} := [v]$ and $i := 1$
2. While $i \leq \text{Length}(\mathcal{B})$ do
3. For j from 1 to k do
4. If $y := \mathcal{B}[i] \cdot A_j \notin \langle \mathcal{B} \rangle_{\mathbb{F}}$ then
5. Append y to the end of $\mathcal{B}$
6. Set $i := i + 1$

Norton's irreducibility criterion

Let $\mathcal{A} = \langle A_1, \ldots, A_k \rangle_{\mathrm{Alg}} \leq \mathbb{F}^{d \times d}$ be a matrix algebra and $B \in \mathcal{A}$ a singular element. Let $\mathcal{A}^t := \langle A_1^t, \ldots, A_k^t \rangle_{\mathrm{Alg}}$.

Norton's irreducibility criterion

Let $\mathcal{A} = \langle A_1, \ldots, A_k \rangle_{\text{Alg}} \leq \mathbb{F}^{d \times d}$ be a matrix algebra and $B \in \mathcal{A}$ a singular element. Let $\mathcal{A}^t := \langle A_1^t, \ldots, A_k^t \rangle_{\text{Alg}}$.

Theorem (Norton)

At least one of the following holds:

1. *There is a* $0 \neq v \in \ker B$ *such that* $v\mathcal{A} \neq V$.
2. *For all* $v \in \ker B^t$ *holds* $v\mathcal{A}^t \neq V$.
3. *The natural module* $V := \mathbb{F}^{1 \times d}$ *is irreducible.*

Norton's irreducibility criterion

Let $\mathcal{A} = \langle A_1, \ldots, A_k \rangle_{\mathrm{Alg}} \leq \mathbb{F}^{d \times d}$ be a matrix algebra and $B \in \mathcal{A}$ a singular element. Let $\mathcal{A}^t := \langle A_1^t, \ldots, A_k^t \rangle_{\mathrm{Alg}}$.

Theorem (Norton)

At least one of the following holds:

1. *There is a $0 \neq v \in \ker B$ such that $v\mathcal{A} \neq V$.*
2. *For all $v \in \ker B^t$ holds $v\mathcal{A}^t \neq V$.*
3. *The natural module $V := \mathbb{F}^{1 \times d}$ is irreducible.*

Proof: Assume that 1 and 3 do not hold, so there is an invariant subspace $0 < W < V$, say of dimension e.

Norton's irreducibility criterion

Let $\mathcal{A} = \langle A_1, \ldots, A_k \rangle_{\mathrm{Alg}} \le \mathbb{F}^{d \times d}$ be a matrix algebra and $B \in \mathcal{A}$ a singular element. Let $\mathcal{A}^t := \langle A_1^t, \ldots, A_k^t \rangle_{\mathrm{Alg}}$.

Theorem (Norton)

At least one of the following holds:

1. *There is a* $0 \ne v \in \ker B$ *such that* $v\mathcal{A} \ne V$.
2. *For all* $v \in \ker B^t$ *holds* $v\mathcal{A}^t \ne V$.
3. *The natural module* $V := \mathbb{F}^{1 \times d}$ *is irreducible.*

Proof: Assume that 1 and 3 do not hold, so there is an invariant subspace $0 < W < V$, say of dimension e.

We can now choose a basis $(w_1, \ldots, w_e)$ of W and extend it to a basis $(w_1, \ldots, w_e, v_1, \ldots, v_{d-e})$ of V and write all matrices with respect to this basis.

Norton's irreducibility criterion

Let $\mathcal{A} = \langle A_1, \ldots, A_k \rangle_{\mathrm{Alg}} \le \mathbb{F}^{d \times d}$ be a matrix algebra and $B \in \mathcal{A}$ a singular element. Let $\mathcal{A}^t := \langle A_1^t, \ldots, A_k^t \rangle_{\mathrm{Alg}}$.

Theorem (Norton)

At least one of the following holds:

1. *There is a* $0 \ne v \in \ker B$ *such that* $v\mathcal{A} \ne V$.
2. *For all* $v \in \ker B^t$ *holds* $v\mathcal{A}^t \ne V$.
3. *The natural module* $V := \mathbb{F}^{1 \times d}$ *is irreducible.*

Proof: Assume that 1 and 3 do not hold, so there is an invariant subspace $0 < W < V$, say of dimension e.

We can now choose a basis $(w_1, \ldots, w_e)$ of W and extend it to a basis $(w_1, \ldots, w_e, v_1, \ldots, v_{d-e})$ of V and write all matrices with respect to this basis.

Let $T := (w_1, \ldots, w_e, v_1, \ldots, v_{d-e})$ and $B' := TBT^{-1}$.

Proof of Norton's criterion

Theorem (Norton)

At least one of the following holds:

1. *There is a $0 \neq v \in \ker B$ such that $v\mathcal{A} \neq V$.*
2. *For all $v \in \ker B^t$ holds $v\mathcal{A}^t \neq V$.*
3. *The natural module $V := \mathbb{F}^{1\times d}$ is irreducible.*

Proof cont'd: Now, $B' = TBT^{-1}$ looks like this:

$$B' = \begin{bmatrix} M & 0 \\ * & N \end{bmatrix}, \text{ where } M \in \mathbb{F}^{e\times e}, N \in \mathbb{F}^{(d-e)\times(d-e)}.$$

Proof of Norton's criterion

Theorem (Norton)

At least one of the following holds:

1. *There is a* $0 \neq v \in \ker B$ *such that* $v\mathcal{A} \neq V$.
2. *For all* $v \in \ker B^t$ *holds* $v\mathcal{A}^t \neq V$.
3. *The natural module* $V := \mathbb{F}^{1\times d}$ *is irreducible.*

Proof cont'd: Now, $B' = TBT^{-1}$ looks like this:

$$B' = \begin{bmatrix} M & 0 \\ * & N \end{bmatrix}, \text{ where } M \in \mathbb{F}^{e\times e}, N \in \mathbb{F}^{(d-e)\times(d-e)}.$$

Since 1 does not hold, $\ker B \cap W = \{0\}$.

Proof of Norton's criterion

Theorem (Norton)

At least one of the following holds:

1. *There is a* $0 \neq v \in \ker B$ *such that* $v\mathcal{A} \neq V$*.*
2. *For all* $v \in \ker B^t$ *holds* $v\mathcal{A}^t \neq V$*.*
3. *The natural module* $V := \mathbb{F}^{1\times d}$ *is irreducible.*

Proof cont'd: Now, $B' = TBT^{-1}$ looks like this:

$$B' = \begin{bmatrix} M & 0 \\ * & N \end{bmatrix}, \text{ where } M \in \mathbb{F}^{e\times e}, N \in \mathbb{F}^{(d-e)\times(d-e)}.$$

Since 1 does not hold, $\ker B \cap W = \{0\}$.
Thus M has full rank e.

Proof of Norton's criterion

Theorem (Norton)

At least one of the following holds:

1. *There is a $0 \neq v \in \ker B$ such that $v\mathcal{A} \neq V$.*
2. *For all $v \in \ker B^t$ holds $v\mathcal{A}^t \neq V$.*
3. *The natural module $V := \mathbb{F}^{1\times d}$ is irreducible.*

Proof cont'd: Now, $B' = TBT^{-1}$ looks like this:

$$B' = \begin{bmatrix} M & 0 \\ * & N \end{bmatrix}, \text{ where } M \in \mathbb{F}^{e\times e}, N \in \mathbb{F}^{(d-e)\times(d-e)}.$$

Since 1 does not hold, $\ker B \cap W = \{0\}$.
Thus M has full rank e.
If $\operatorname{rank} B' =: r < d$, then $\operatorname{rank} N = r - e < d - e$.

Proof of Norton's criterion

Theorem (Norton)

At least one of the following holds:

1. *There is a $0 \neq v \in \ker B$ such that $v\mathcal{A} \neq V$.*
2. *For all $v \in \ker B^t$ holds $v\mathcal{A}^t \neq V$.*
3. *The natural module $V := \mathbb{F}^{1 \times d}$ is irreducible.*

Proof cont'd: Now, $B' = TBT^{-1}$ looks like this:

$$B' = \begin{bmatrix} M & 0 \\ * & N \end{bmatrix}, \text{ where } M \in \mathbb{F}^{e \times e}, N \in \mathbb{F}^{(d-e)\times(d-e)}.$$

Since 1 does not hold, $\ker B \cap W = \{0\}$.
Thus M has full rank e.
If $\operatorname{rank} B' =: r < d$, then $\operatorname{rank} N = r - e < d - e$.
Thus $\dim_{\mathbb{F}}(\ker N) = d - r = d - e - \operatorname{rank} N$.

Proof of Norton's criterion

Theorem (Norton)

At least one of the following holds:

1. *There is a* $0 \neq v \in \ker B$ *such that* $v\mathcal{A} \neq V$.
2. *For all* $v \in \ker B^t$ *holds* $v\mathcal{A}^t \neq V$.
3. *The natural module* $V := \mathbb{F}^{1\times d}$ *is irreducible.*

Proof cont'd: Now, $B' = TBT^{-1}$ looks like this:

$$B' = \begin{bmatrix} M & 0 \\ * & N \end{bmatrix}, \text{ where } M \in \mathbb{F}^{e\times e}, N \in \mathbb{F}^{(d-e)\times(d-e)}.$$

Since 1 does not hold, $\ker B \cap W = \{0\}$.
Thus M has full rank e.
If $\operatorname{rank} B' =: r < d$, then $\operatorname{rank} N = r - e < d - e$.
Thus $\dim_{\mathbb{F}}(\ker N) = d - r = d - e - \operatorname{rank} N$.
Now consider $B'^t = (T^t)^{-1} B^t T^t$:
$\ker B'^t$ is contained in an $(T\mathcal{A}T^{-1})^t$-invariant subspace.

Proof of Norton's criterion

Theorem (Norton)

At least one of the following holds:

1. *There is a* $0 \neq v \in \ker B$ *such that* $v\mathcal{A} \neq V$.
2. *For all* $v \in \ker B^t$ *holds* $v\mathcal{A}^t \neq V$.
3. *The natural module* $V := \mathbb{F}^{1\times d}$ *is irreducible.*

Proof cont'd: Now, $B' = TBT^{-1}$ looks like this:

$$B' = \begin{bmatrix} M & 0 \\ * & N \end{bmatrix}, \text{ where } M \in \mathbb{F}^{e\times e}, N \in \mathbb{F}^{(d-e)\times(d-e)}.$$

Since 1 does not hold, $\ker B \cap W = \{0\}$.
Thus M has full rank e.
If $\operatorname{rank} B' =: r < d$, then $\operatorname{rank} N = r - e < d - e$.
Thus $\dim_{\mathbb{F}}(\ker N) = d - r = d - e - \operatorname{rank} N$.
Now consider $B'^t = (T^t)^{-1} B^t T^t$:
$\ker B'^t$ is contained in an $(T\mathcal{A}T^{-1})^t$-invariant subspace.
Thus $\ker B^t$ is contained in an $\mathcal{A}^t$-invariant subspace. ∎

Chopping modules I

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.

Chopping modules I

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
"Chopping" means computing a composition series.

Chopping modules I

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
"Chopping" means computing a composition series.
The MeatAxe basically does the following:

Chopping modules I

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
"Chopping" means computing a composition series.
The MeatAxe basically does the following:

A basic step of "Chop"

1. Find an element $B \in \mathcal{A}$ with small, non-trivial kernel

Chopping modules I

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
"Chopping" means computing a composition series.
The MeatAxe basically does the following:

A basic step of "Chop"

1. Find an element $B \in \mathcal{A}$ with small, non-trivial kernel
2. Compute $\ker B$

Chopping modules I

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
"Chopping" means computing a composition series.
The MeatAxe basically does the following:

A basic step of "Chop"

1. Find an element $B \in \mathcal{A}$ with small, non-trivial kernel
2. Compute $\ker B$
3. Spinup all $0 \neq v \in \ker B$

Chopping modules I

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
"Chopping" means computing a composition series.
The MeatAxe basically does the following:

A basic step of "Chop"

1. Find an element $B \in \mathcal{A}$ with small, non-trivial kernel
2. Compute $\ker B$
3. Spinup all $0 \neq v \in \ker B$
4. If some $v\mathcal{A} < V$, we found a submodule, goto 7

Chopping modules I

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
"Chopping" means computing a composition series.
The MeatAxe basically does the following:

A basic step of "Chop"

1. Find an element $B \in \mathcal{A}$ with small, non-trivial kernel
2. Compute $\ker B$
3. Spinup all $0 \neq v \in \ker B$
4. If some $v\mathcal{A} < V$, we found a submodule, goto 7
5. Otherwise spinup one $0 \neq v \in \ker B^t$ under $\mathcal{A}^t$

Chopping modules I

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
"Chopping" means computing a composition series.
The MeatAxe basically does the following:

A basic step of "Chop"

1. Find an element $B \in \mathcal{A}$ with small, non-trivial kernel
2. Compute $\ker B$
3. Spinup all $0 \neq v \in \ker B$
4. If some $v\mathcal{A} < V$, we found a submodule, goto 7
5. Otherwise spinup one $0 \neq v \in \ker B^t$ under $\mathcal{A}^t$
6. If $v\mathcal{A}^t = V$, we have proved V to be irreducible, stop

Chopping modules I

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
"Chopping" means computing a composition series.
The MeatAxe basically does the following:

A basic step of "Chop"

1. Find an element $B \in \mathcal{A}$ with small, non-trivial kernel
2. Compute $\ker B$
3. Spinup all $0 \neq v \in \ker B$
4. If some $v\mathcal{A} < V$, we found a submodule, goto 7
5. Otherwise spinup one $0 \neq v \in \ker B^t$ under $\mathcal{A}^t$
6. If $v\mathcal{A}^t = V$, we have proved V to be irreducible, stop
7. If $0 < W < V$ is invariant, compute action on W and V/W and recurse (with smaller dimensions!)

Chopping modules II

The result of "Chop" is a composition series

$$\{0\} = V_{\ell+1} < V_\ell < V_{\ell-1} < \cdots < V_1 = V$$

such that all V_j/V_{j+1} are irreducible.

Chopping modules II

The result of "Chop" is a composition series

$$\{0\} = V_{\ell+1} < V_\ell < V_{\ell-1} < \cdots < V_1 = V$$

such that all V_j/V_{j+1} are irreducible.

Actually, we find a base change $T \in \mathbb{F}^{d\times d}$, such that all matrices TA_iT^{-1} for $1 \leq i \leq k$ look like this:

$$TA_iT^{-1} = \begin{bmatrix} M_\ell^{(i)} & 0 & \cdots & 0 \\ * & M_{\ell-1}^{(i)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ * & \cdots & * & M_1^{(i)} \end{bmatrix}$$

and the matrices $M_j^{(i)}$ describe the action of $\mathcal{A}$ on V_j/V_{j+1}.

Chopping modules II

The result of "Chop" is a composition series

$$\{0\} = V_{\ell+1} < V_\ell < V_{\ell-1} < \cdots < V_1 = V$$

such that all V_j/V_{j+1} are irreducible.

Actually, we find a base change $T \in \mathbb{F}^{d \times d}$, such that all matrices TA_iT^{-1} for $1 \leq i \leq k$ look like this:

$$TA_iT^{-1} = \begin{bmatrix} M_\ell^{(i)} & 0 & \cdots & 0 \\ * & M_{\ell-1}^{(i)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ * & \cdots & * & M_1^{(i)} \end{bmatrix}$$

and the matrices $M_j^{(i)}$ describe the action of $\mathcal{A}$ on V_j/V_{j+1}.

A more detailed analysis shows that the MeatAxe can identify isomorphism types of irreducible modules.

Overview over available algorithms

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1 \times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d \times d}$.
The MeatAxe can do the following for you:

Overview over available algorithms

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
The MeatAxe can do the following for you:

- Compute a composition series.

Overview over available algorithms

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
The MeatAxe can do the following for you:

- Compute a composition series.
- Find homomorphism spaces from an irreducible module to another one.

Overview over available algorithms

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1 \times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d \times d}$.
The MeatAxe can do the following for you:

- Compute a composition series.
- Find homomorphism spaces from an irreducible module to another one.
- Identify the isomorphism type of irreducible modules.

Overview over available algorithms

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1 \times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d \times d}$.
The MeatAxe can do the following for you:

- Compute a composition series.
- Find homomorphism spaces from an irreducible module to another one.
- Identify the isomorphism type of irreducible modules.
- Compute the socle and radical series.

Overview over available algorithms

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1 \times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d \times d}$.
The MeatAxe can do the following for you:

- Compute a composition series.
- Find homomorphism spaces from an irreducible module to another one.
- Identify the isomorphism type of irreducible modules.
- Compute the socle and radical series.
- Compute the submodule lattice.

Overview over available algorithms

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1\times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d\times d}$.
The MeatAxe can do the following for you:

- Compute a composition series.
- Find homomorphism spaces from an irreducible module to another one.
- Identify the isomorphism type of irreducible modules.
- Compute the socle and radical series.
- Compute the submodule lattice.
- Compute homomorphism spaces between arbitrary modules.

Overview over available algorithms

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1 \times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d \times d}$.
The MeatAxe can do the following for you:

- Compute a composition series.
- Find homomorphism spaces from an irreducible module to another one.
- Identify the isomorphism type of irreducible modules.
- Compute the socle and radical series.
- Compute the submodule lattice.
- Compute homomorphism spaces between arbitrary modules.
- Compute cohomology groups.

Overview over available algorithms

Assume we are given an $\mathcal{A}$-module $V = \mathbb{F}^{1 \times d}$ by matrices $A_1, \ldots, A_k \in \mathbb{F}^{d \times d}$.
The MeatAxe can do the following for you:

- Compute a composition series.
- Find homomorphism spaces from an irreducible module to another one.
- Identify the isomorphism type of irreducible modules.
- Compute the socle and radical series.
- Compute the submodule lattice.
- Compute homomorphism spaces between arbitrary modules.
- Compute cohomology groups.
- Compute condensed modules.

The End

Bibliography

Derek F. Holt and Sarah Rees.
Testing modules for irreducibility.
J. Austral. Math. Soc. Ser. A, 57(1):1–16, 1994.

Gábor Ivanyos and Klaus Lux.
Treating the exceptional cases of the MeatAxe.
Experiment. Math., 9(3):373–381, 2000.

R. A. Parker.
The computer calculation of modular characters (the meat-axe).
In *Computational group theory (Durham, 1982)*, pages 267–274. Academic Press, London, 1984.