

# Pflichtenheft

<b>Projektbezeichnung</b>	Abschlussprojekt Gauß-Algorithmus
<b>Auftraggeber</b>	Prof. Dr. Paul Kirchberg
<b>Auftragnehmer</b>	Judith Romer, Timo Heiß, Nick Hillebrand, Tom Zehle
<b>Erstellt am</b>	20.02.2023
<b>Letzte Änderung am</b>	13.08.2023
<b>Aktuelle Version</b>	1.2

## Änderungsverlauf

Nr.	Datum	Version	Änderungen	Autor
1	20.02.2023	1.0	Erstellung	Timo Heiß, Nick Hillebrand
2	20.03.2023	1.1	- Anpassung 2.3 Zielgruppe - Anpassung F062 - Anpassung 4. Testung - Anpassung Zeitplan in 7. Vorgehen	Timo Heiß, Nick Hillebrand
3	13.08.2023	1.2	- Anpassung 1.3 Budget in Vorlesungseinheiten - Aufnahme funktionale ANF F065 Simplex-Algorithmus - Aufnahme funktionale ANF F066 Ein-/Ausgabe leeren - Anpassung von funktionaler ANF F020 - Anpassung 6.2 Produktdokumentation - Anpassung 6.3 Projektdokumentation - Anpassung 7 Vorgehen	Timo Heiß, Nick Hillebrand

# Inhaltsverzeichnis

1 Allgemeines .....	3
1.1 Ziel und Zweck dieses Dokuments.....	3
1.2 Projektbezug .....	3
1.3 Budget.....	3
2 Rahmenbedingungen und Produkteinsatz.....	3
2.1 Zielbestimmung .....	3
2.2 Anwendungsbereich .....	3
2.3 Zielgruppen .....	3
2.4 Produktumgebung / Betriebsbedingungen.....	3
3 Produktfunktionen / Anforderungen .....	4
3.1 Funktionale Anforderungen .....	4
3.2 Nichtfunktionale Anforderungen.....	7
4 Testung .....	7
5 Monitoring / Support nach Übergabe .....	7
6 Dokumentation .....	8
6.1 Anwenderdokumentation.....	8
6.2 Produktdokumentation.....	8
6.3 Projektdokumentation .....	8
7 Vorgehen .....	9
8 Abnahme .....	9
9 Genehmigung .....	10
10 Anhang .....	11
10.1 Projektausschreibung .....	11
10.2 Oberflächenprototypen.....	12

# 1 Allgemeines

## 1.1 Ziel und Zweck dieses Dokuments

Dieses Dokument dient als bindende Vereinbarung zwischen dem Auftraggeber und den Auftragnehmern. Für das Abschlussprojekt „Gauß-Algorithmus“ werden hier alle Anforderungen an die Software-Lösung beschrieben und wie diese umzusetzen sind.

## 1.2 Projektbezug

Dieses Pflichtenheft bezieht sich auf das Abschlussprojekt mit dem Titel „Gauß-Algorithmus“ im 5. und 6. Semester des Kurses WWIDS120 an der DHBW Ravensburg.

## 1.3 Budget

Für die Umsetzung des Projekts „Gauß-Algorithmus“ stehen je Auftragnehmer 210 Stunden (à 45min) zur Verfügung. Bei vier Personen sind dies  $4 \times 210 = 840$ . Dieses Budget ist weitestgehend auszureizen und nicht zu überschreiten.

# 2 Rahmenbedingungen und Produkteinsatz

## 2.1 Zielbestimmung

Es soll eine Web-Oberfläche mit HTML, CSS und JavaScript erstellt werden. In der Web-Oberfläche soll mit dem Gauß-Algorithmus Schritt für Schritt eine einzugebende Matrizenberechnung gelöst werden können. Dabei sind die einzelnen Zeilenoperationen durch den Anwender selbst bestimmbar. Beim Einsatz der Anwendung soll ein Verständnis des Algorithmus beim Anwender geschaffen werden (vgl. Anhang 10.1 Projektausschreibung).

## 2.2 Anwendungsbereich

Die Web-Oberfläche soll in der Vorlesung „Analysis & Lineare Algebra“ im ersten Studienjahr an der DHBW Ravensburg eingesetzt werden. Dabei soll den Studierenden ein Verständnis für den Gauß-Algorithmus bei der Lösung von Gleichungssystemen und bei der Lösung von Matrizengleichungen bzw. bei der Inversenberechnung vermittelt werden, während gleichzeitig der Schreibaufwand bei der sonst sehr schreibintensiven Berechnung wegfällt.

## 2.3 Zielgruppen

Die Web-Oberfläche wird von Studierenden und Schülern im Lernprozess genutzt.

## 2.4 Produktumgebung / Betriebsbedingungen

Die Web-Oberfläche kann mithilfe eines Webbrowsers aufgerufen werden. Dieser muss HTML5, CSS sowie die aktuelle JavaScript-Version (2022) unterstützen. Die HTML-, CSS und JS-Dateien müssen auf einem Webserver gehostet werden, damit die Anwendung laufen kann. Alternativ können die Dateien über ein Filesystem aufgerufen werden. Dabei muss der Webbrowser aber mit Flag `--allow-file-access-from-files` gestartet werden.

## 3 Produktfunktionen / Anforderungen

### 3.1 Funktionale Anforderungen

ID	Anforderung	Beschreibung	Art
F010	Startseite mit Navigation	Der Nutzer soll beim Start der Anwendung auf eine Startseite geführt werden, über die er auf die übrigen Funktionalitäten zugreifen kann.	Muss <sup>1</sup>
F020	Matrizenrechnungen durchführen	Der Nutzer soll bis zu 26 eingegebene Matrizen (Bezeichnungen A-Z) beliebiger Größe (min. 2x2, max. 9x9) addieren, subtrahieren und multiplizieren, mit Skalaren multiplizieren und Determinanten und Inversen berechnen können in beliebiger Kombination, wobei das Ergebnis von der Anwendung ermittelt wird.	Muss
F030	Matrixoperationen ausführen	Der Nutzer soll die Möglichkeit haben, sich für eine eingegebene Matrix beliebiger Größe (min. 2x2, max. 9x9) die Inverse, die Determinante und die transponierte Matrix von der Anwendung berechnen und ausgeben zu lassen.	Muss
F040	Gleichungssystem schrittweise lösen	Der Nutzer soll ein Gleichungssystem mit beliebiger Anzahl Gleichungen, Variablen und rechten Seiten eingeben können. Dieses soll der Nutzer mithilfe der Anwendung schrittweise lösen können.	Muss
F041	Zeilenoperationen für Gleichungssystem durchführen	Zum Lösen des Gleichungssystems soll der Nutzer Zeilenoperationen über eine Reihe an Combo-Boxen eingeben können. Das Ergebnis der Zeilenoperationen soll von der Anwendung berechnet werden und vom Nutzer weiterverwendet werden können.	Muss
F042	Anzahl Lösungen des Gleichungssystems angeben	Die Anwendung soll die Anzahl an Lösungen des Gleichungssystems (genau eine, keine, unendlich viele) erkennbar darstellen.	Muss
F043	Ungültige Zeilenoperationen sperren	Zeilenoperationen, die den Lösungsraum verändern, sollen von der Anwendung nicht zugelassen werden.	Muss
F045	Gleichungssystem lösen lassen	Der Nutzer soll die Möglichkeit haben, die Endlösung des Gleichungssystems von der Anwendung berechnen zu lassen.	Muss
F050	Matrizengleichung / Matrixinversion schrittweise durchführen	Der Nutzer soll eine Matrizengleichung aus zwei Matrizen mit beliebiger gleicher Dimensionalität (min. 2, max. 9) eingeben können. Er soll die Matrizengleichung selbstständig mithilfe der Anwendung schrittweise lösen können. Da hiermit auch eine Matrix invertiert werden kann, soll die zweite Matrix standardmäßig mit der Einheitsmatrix vorbelegt sein.	Muss

<sup>1</sup> Zusätzlich zur Unterscheidung in funktionale und nichtfunktionale Anforderungen, werden in der Spalte „Art“ die Anforderungen weiter in folgende Kategorien eingeteilt:

- **Muss:** Die Anforderung muss in jedem Fall umgesetzt werden.
- **Kann:** Die Anforderung soll, sofern dies mit dem Budget möglich ist, umgesetzt werden. Dabei wird eine Priorisierung von 1-3 vorgenommen, wobei 1 zuerst, 3 zuletzt umzusetzen ist.
- **Abgrenzung:** Die Anforderung dient der Abgrenzung und beschreibt etwas, das explizit nicht umgesetzt werden soll.

F051	Zeilenoperationen für Matrizengleichung / Matrixinversion durchführen	Zur Lösung der Matrizengleichung bzw. Matrixinversion soll der Nutzer Zeilenoperationen über eine Reihe an Combo-Boxen eingeben können. Das Ergebnis der Zeilenoperationen soll von der Anwendung berechnet und vom Nutzer weiterverwendet werden können.	Muss
F052	Lösung der Matrizengleichung / Inversenberechnung durchführen lassen	Der Nutzer soll die Möglichkeit haben, die komplette Lösung der Matrizengleichung bzw. Matrixinversion von der Anwendung berechnen zu lassen.	Muss
F060	Fehlermeldungen ausgeben	Die Anwendung soll für vom Nutzer verursachte Probleme (z.B. Matrix mit linearen Abhängigkeiten invertieren) entsprechende Fehlermeldungen ausgeben.	Muss
F061	Bedienungshilfe stellen	Der Nutzer soll bei Problemen auf eine Bedienungshilfe zugreifen können (Gesamt und individuell pro Seite), um eine Bedienung ohne Hilfe von außen sicherzustellen.	Muss
F062	Keine Eingabe/Ausgabe entspricht 0	Gibt der Nutzer keine Zahl in ein Feld einer Matrix ein, so ist dies als 0 zu interpretieren. Ebenso werden 0 Felder einer Matrix als leere Zelle dargestellt.	Muss
F063	Navigation mit Pfeiltasten	Der Nutzer soll zwischen den Eingabefeldern der Web-Oberfläche mit Pfeiltasten einfach navigieren können.	Muss
F064	Brüche und Dezimalzahlen eingeben und darstellen	Der Nutzer soll Zahlen als Ganzzahlen, Brüche oder Dezimalzahlen eingeben können. Bei der Darstellung des Ergebnisses soll der Anwender die Form (Brüche oder Dezimalzahlen) flexibel wählen können.	Muss
F065	Lineares Optimierungsproblem mit Simplex Algorithmus üben	Der Nutzer soll ein lineares Optimierungsproblem mit Nebenbedingungen im Standardformat (Minimierung der Zielfunktion, nur Gleichheitsnebenbedingungen, nur nicht-negative Variablen, rechte Seiten der Nebenbedingungen nicht-negativ) in eine Simplex-Tabelle eingeben können, das dann vom System mithilfe des Simplex-Algorithmus gelöst wird.	Muss
F066	Ein- und Ausgabefelder leeren	Der Nutzer soll die Möglichkeit haben, alle Matrizen (Ein- und Ausgabefelder) zu leeren.	Muss
F070	Historie für die Zeilenoperationen	Der Nutzer soll die von ihm durchgeführten Zeilenoperationen beim Lösen des Gleichungssystems sowie bei der Inversenberechnung in einer Historie einsehen können.	Kann, Priorität 1
F071	Ein-/Ausblenden der Zeilenoperationen	Der Nutzer soll die Comboboxen für die Zeilenoperationen beim Gleichungssystem und bei der Inversenberechnung ein- und ausblenden können.	Kann, Priorität 2

F072	Tipp für Zeilenoperationen einholen	Der Nutzer soll die Möglichkeit haben, für den nächsten Schritt zur Erzeugung der Stufenform die Zeilenoperation-Comboboxen von der Anwendung befüllen zu lassen (als „Tipp“).	Kann, Priorität 2
F073	Übungsaufgaben laden	Der Nutzer soll die Möglichkeit haben, für das Gleichungssystem und die Inversenberechnung Übungsaufgaben (eingeteilt in drei Schwierigkeitslevel) laden zu können.	Kann, Priorität 2
F074	Historie exportieren/drucken	Der Nutzer soll die Möglichkeit haben, die Historie für die Zeilenoperationen beim Gleichungssystem und bei der Inversenberechnung als PDF zu exportieren und zu drucken.	Kann, Priorität 3

### 3.2 Nichtfunktionale Anforderungen

ID	Anforderung	Beschreibung	Art
NF010	HTML, CSS, JS	Die Anwendung ist als Web-Oberfläche mit HTML, CSS und JS umzusetzen.	Muss
NF011	Keine Frameworks & Bibliotheken	HTML und JS sind ohne ergänzende Frameworks und Bibliotheken zu verwenden.	Abgrenzung
NF012	Kein Hosting	Es müssen lediglich die HTML-, CSS- und JS-Dateien erstellt werden. Deren Hosting ist nicht inbegriffen.	Abgrenzung
NF020	Internes Rechnen mit Brüchen	Die Anwendung soll intern mit Brüchen rechnen.	Muss
NF021	Interne Berechnungen über Gauß-Algorithmus	Die Anwendung soll intern zur Berechnung der Determinante sowie zur Endlösung des Gleichungssystems und Inversenberechnung den Gauß-Algorithmus anwenden.	Muss
NF022	Interne Berechnungen mit Simplex-Algorithmus	Die Anwendung soll intern auch mit dem Simplex-Algorithmus Gleichungssysteme und Inversenberechnungen lösen können.	
NF030	Ansprechendes Design	Die Web-Oberfläche soll ein ansprechendes Design aufweisen.	Muss
NF031	Responsives Design	Die Grundgrößen Laptop/PC und Tablet sollen problemlos angezeigt werden können (Muss-Anforderung). Andere Größen wie Smartphones sollen wenn möglich ebenfalls unterstützt werden (Kann-Anforderung, Priorität 2).	Muss / Kann, Priorität 2

## 4 Testung

Die Grundfunktionen sollen mithilfe von Unit-Tests abgedeckt werden. Hauptziel hiervon ist es, die Wartbarkeit und Erweiterbarkeit zu gewährleisten. Es wird von Entwickler-Seite eine Test-Coverage von 75% angestrebt. Dabei handelt es sich um einen in der Praxis gängigen Anteil. Bei höherer Test-Coverage stünde der Grenznutzen nicht mehr im Verhältnis zum Aufwand.

Außerdem sollen Integrationstests durchgeführt werden, bei denen das Zusammenspiel verschiedener Komponenten im Vordergrund steht.

Automatisierte Tests sind durch manuelle UI-Tests sowie Performance Tests für die größtmöglichen Berechnungen zu ergänzen. Im Rahmen der finalen Abnahme ist ein Abnahmetest gemeinsam mit dem Kunden durchzuführen. Entsprechende Testfälle sind zu dokumentieren.

## 5 Monitoring / Support nach Übergabe

Die Zuständigkeit der Auftragnehmer endet mit Übergabe der HTML-, CSS- und JS-Dateien sowie von Projekt- und Produktdokumentation im Zuge einer Abschlusspräsentation. Ein Support über diesen Zeitpunkt hinaus findet nicht statt.

## 6 Dokumentation

### 6.1 Anwenderdokumentation

Alle Hinweise bezüglich der Bedienung sind in der Anwendung selbst integriert (siehe Anforderungen). Eine separate Dokumentation für den Anwender wird daher nicht erstellt.

### 6.2 Produktdokumentation

Dem Auftraggeber wird eine Produktdokumentation zur Verfügung gestellt. Diese umfasst den allgemeinen Aufbau der Anwendung (Ordnerstruktur, Dateien, etc.), ebenso wie die aus JS-Doc-Kommentaren resultierende Dokumentation für Klassen, Methoden, Instanzvariablen und Funktionen. Weiter wird ein UML-Klassendiagramm beigelegt. Darüber hinaus beinhaltet die Produktdokumentation die Oberflächenprototypen<sup>2</sup>. Eine kurze Beschreibung der Funktionalitäten der Anwendung wird ebenfalls beigelegt. Zudem werden die Betriebsbedingungen spezifiziert und eine Dokumentation der Testfälle angelegt.

### 6.3 Projektdokumentation

Ebenso wird eine Projektdokumentation erstellt. Diese beinhaltet Details rund um das Projektmanagement. Neben einer Beschreibung der grundsätzlichen Methodik, umfasst die Projektdokumentation eine Zeit-/Aufwandserfassung, die Rollenverteilung im Team, eine Roadmap, eine Beschreibung der Vorgehensweise bei der Produktentwicklung sowie eine kurze Retrospektive. Im Anhang werden die Meeting-Protokolle aus den Terminen mit dem Auftraggeber sowie dieses Pflichtenheft angelegt.

---

<sup>2</sup> Die Oberflächenprototypen sind außerdem im Anhang dieses Pflichtenhefts zu finden.



## 7 Vorgehen

Beim Kick-Off werden zunächst die Anforderungen im Rahmen der integrativen, agilen Vorgehensweise gemeinsam erarbeitet. Auf deren Basis wird dieses Pflichtenheft sowie erste Oberflächenprototypen erstellt. Darauf aufbauend werden die Prototypen in Quellcode umgesetzt. Zunächst wird ein MVP (Minimum Viable Product) erstellt, der sämtliche Muss-Anforderungen berücksichtigt. Anschließend wird das Feedback aus dem MVP eingearbeitet, die Funktionalitäten überarbeitet, so dass V1 (Version 1) alle Muss-Anforderungen in getesteter Form beinhaltet und ggf. erste Kann-Anforderungen umsetzt. Die Produktversion V2 berücksichtigt dann die neu hinzugefügte Muss-Anforderung des Simplex (F065) sowie das Kundenfeedback aus V1 und setzt ggf. weitere Kann-Anforderungen um.

### Roadmap:



## 8 Abnahme

Die Abnahme erfolgt im Rahmen einer Abschlusspräsentation am Ende des 6. Semesters. In diesem Zuge werden die HTML-, CSS- und JS-Dateien übergeben, die die Web-Oberfläche bilden. Darüber hinaus werden die in diesem Dokument genannte Produkt- sowie Projektdokumentation übergeben. Abnahmekriterien für die Web-Oberfläche stellen die in diesem Pflichtenheft spezifizierten Anforderungen dar.

## 9 Genehmigung

### Auftraggeber:

---

Ort, Datum

---

Unterschrift

### Auftragnehmer:

---

Ort, Datum

---

Unterschrift

---

Ort, Datum

---

Unterschrift

---

Ort, Datum

---

Unterschrift

---

Ort, Datum

---

Unterschrift

# 10 Anhang

## 10.1 Projektausschreibung

### GAUß-ALGORITHMUS

#### ■ Problemstellung

In der linearen Algebra wird Gauß-Algorithmus beispielsweise zur Lösung linearer Gleichungssysteme, für Matrizengleichungen und zur Determinantenberechnung eingesetzt. Der Algorithmus arbeitet iterativ auf Matrizendarstellungen, an denen sich in jedem Schritt nur wenig ändert, aber viel zu schreiben ist. Dies lässt sich über eine passende webbasierte Anwendung vereinfachen.

#### ■ Ziel des Projekts

- Implementierung einer webbasierten Oberfläche, in der der Gauß-Algorithmus Schritt für Schritt eine einzugebende Matrizenberechnung löst. Die durchzuführenden Zeilenoperationen sind hierbei in jedem Schritt vom Anwender selbst bestimmbar.
- Die Oberfläche ist so zu gestalten, dass beim Einsatz ein Verständnis des Algorithmus geschaffen wird.

#### ■ Rahmenbedingungen

- Die Anwendung ist als Web-Anwendung umzusetzen.
- HTML und JavaScript ohne ergänzende Frameworks/Bibliotheken
- Kanban im Projektmanagement

#### ■ Kunde

Prof. Dr. Kirchberg

## 10.2 Oberflächenprototypen

Startseite

Matrizenrechnung

Matrixoperationen

GI-System

Inversenberechnung

### Startseite

Herzlichen Willkommen auf dieser Seite.  
Folgende Inhalte stehen Dir hier zur Verfügung:

- [Matrizenrechnung](#)
- [Matrixoperationen](#)
- [GI-System](#)
- [Inversenberechnung](#)

Auf der Seite [Hilfe](#) findest Du weitere Erklärungen.

Matrizenrech...

https://www.default.com/matrizenrechnungen

Startseite

Matrizenrechnung

Matrixoperationen

GI-System

Inversenberechnung

### Matrizenrechnungen

Hilfe

A =

3

3

3

3

3

3

3

3

3

B =

3

3

3

3

3

3

3

3

3

+ Neue Matrix

Rechnung

Berechnen

A x B + B

=

3

3

3

3

3

3

3

3

3

Matrixopera...

https://www.default.com/Matrixoperationen

Startseite

Matrizenrechnung

Matrixoperationen

GI-System

Inversenberechnung

### Matrixoperationen

Hilfe

Anzahl Zeilen: 

3

Anzahl Spalten: 

3

M<sup>A</sup>(-1)

|M|

M<sup>A</sup>T

Ergebnis

GL-System

<https://www.default.com/gl-system>

[Startseite](#)
[Matrizenberechnung](#)
[Matrixoperationen](#)
[GI-System](#)
[Inversenberechnung](#)

## Gleichungssystem Lösen

Anzahl Gleichungen: 3
Anzahl Variablen: 3
Anzahl rechte Seiten: 1

	$x_1$	$x_2$	$x_3$	$b_i$
(1)	3	3	3	3
(2)	3	3	3	3
(3)	3	3	3	3

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

Tipp
Lösen

Berechnen
Hoch

Berechnung
Historie

	$x_1$	$x_2$	$x_3$	$b_i$
(1)	3	3	3	3
(2)	3	3	3	3
(3)	3	3	3	3

GL-System

<https://www.default.com/gl-system>

[Startseite](#)
[Matrizenberechnung](#)
[Matrixoperationen](#)
[GI-System](#)
[Inversenberechnung](#)

## Gleichungssystem Lösen

Anzahl Gleichungen: 3
Anzahl Variablen: 3
Anzahl rechte Seiten: 1

	$x_1$	$x_2$	$x_3$	$b_i$
(1)	3	3	3	3
(2)	3	3	3	3
(3)	3	3	3	3

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

Tipp
Lösen

Berechnen
Hoch

Berechnung
Historie

Drucken

Historie leeren

### Schritt 1

	$x_1$	$x_2$	$x_3$	$b_i$
(1)	3	3	3	3
(2)	3	3	3	3
(3)	3	3	3	3

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

GL-System

<https://www.default.com/gl-system>

[Startseite](#)
[Matrizenberechnung](#)
[Matrixoperationen](#)
[GI-System](#)
[Inversenberechnung](#)

## Gleichungssystem Lösen

Anzahl Gleichungen: 3

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

Tipp
Lösen

Berechnen
Hoch

Berechnung
Historie

Drucken

Historie leeren

### Schritt 1

	$x_1$	$x_2$	$x_3$	$b_i$
(1)	3	3	3	3
(2)	3	3	3	3
(3)	3	3	3	3

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

↔

1

+

↔

0

↔

(1)

↔

Hilfe - Gleichungssystem Lösen

Bedienungsanleitung

**Matrizengröße Anpassen**  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

**Zeilenoperationen durchführen**  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

**Berechnung ausführen**  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Schließen

Inversenber...

https://www.default.com/matrixinversion

Startseite

Matrizenberechnung

Matrixoperationen

GI-System

Inversenberechnung

Matrixinversion - Matrizengleichung

Hilfe

Matrixdimension: 3

(1)	3	3	3	1			•▼	1	+▼	0	•▼	(1)▼	⬅	Tipp
(2)	3	3	3		1		•▼	1	+▼	0	•▼	(1)▼	⬅	Lösen
(3)	3	3	3			1	•▼	1	+▼	0	•▼	(1)▼	⬅	

Berechnen

Hoch

Berechnung

Historie

(1)	3	3	3	3	3	3
(2)	3	3	3	3	3	3
(3)	3	3	3	3	3	3