

# Entfaltungsprogramm RUN

Natalie Milke

Schule für Astroteilchenphysik 2008  
Obertrubach-Bärnfels

# Inhalt

- Entfaltungsmethode
- Entfaltungsprogramm RUN
- Geplante Verbesserung von RUN
- Optimierungsalgorithmus von V.Curtef
- Zusammenfassung und Ausblick

# Problemstellung

- Teilchen verursachen Signale im Detektor
    - ➔ Eigenschaft des Teilchens (z.B. Energie)
  - Unvollkommenheit des Detektors
    - Verschmieren der Signale
    - Akzeptanz  $< 1$
    - gemessen wird evtl. andere Größe als gesucht
- ➔ Verteilung der gesuchten Größe muss rekonstruiert werden

# Problemstellung

- Zusammenhang zwischen gemessener und gesuchter Verteilung

$$g(y) = \int_a^b A(x, y) f(x) dx$$

- Fredholm-Integral I. Art

# Problemstellung

- Zusammenhang zwischen gemessener und gesuchter Verteilung

$$\underbrace{g(y)}_{\text{gemessen}} = \int_a^b \underbrace{A(x, y)}_{\text{Detektor- einfluss}} \underbrace{f(x)}_{\text{gesucht}} dx$$

- Fredholm-Integral I. Art

# Problemstellung

- Messung liefert diskrete Werte (Histogramme)  
→ Diskretisieren der Integralgleichung:

$$\mathbf{g} = \mathbf{A}\mathbf{f}$$

Darstellung aus Vektoren  $\mathbf{g}$ ,  $\mathbf{f}$  und der Matrix  $\mathbf{A}$

$$g_i = \sum_j A_{ij} f_j$$

# Entfaltung

- Bestimmung von  $f$  durch Umkehren der Matrix-Gleichung
  - ➔ inverses Problem
  - ➔ liefert oft oszillierende Lösungen
  - ➔ Glätten der Lösung nötig: Regularisierung (z.B. Krümmung der Lösung optimieren)

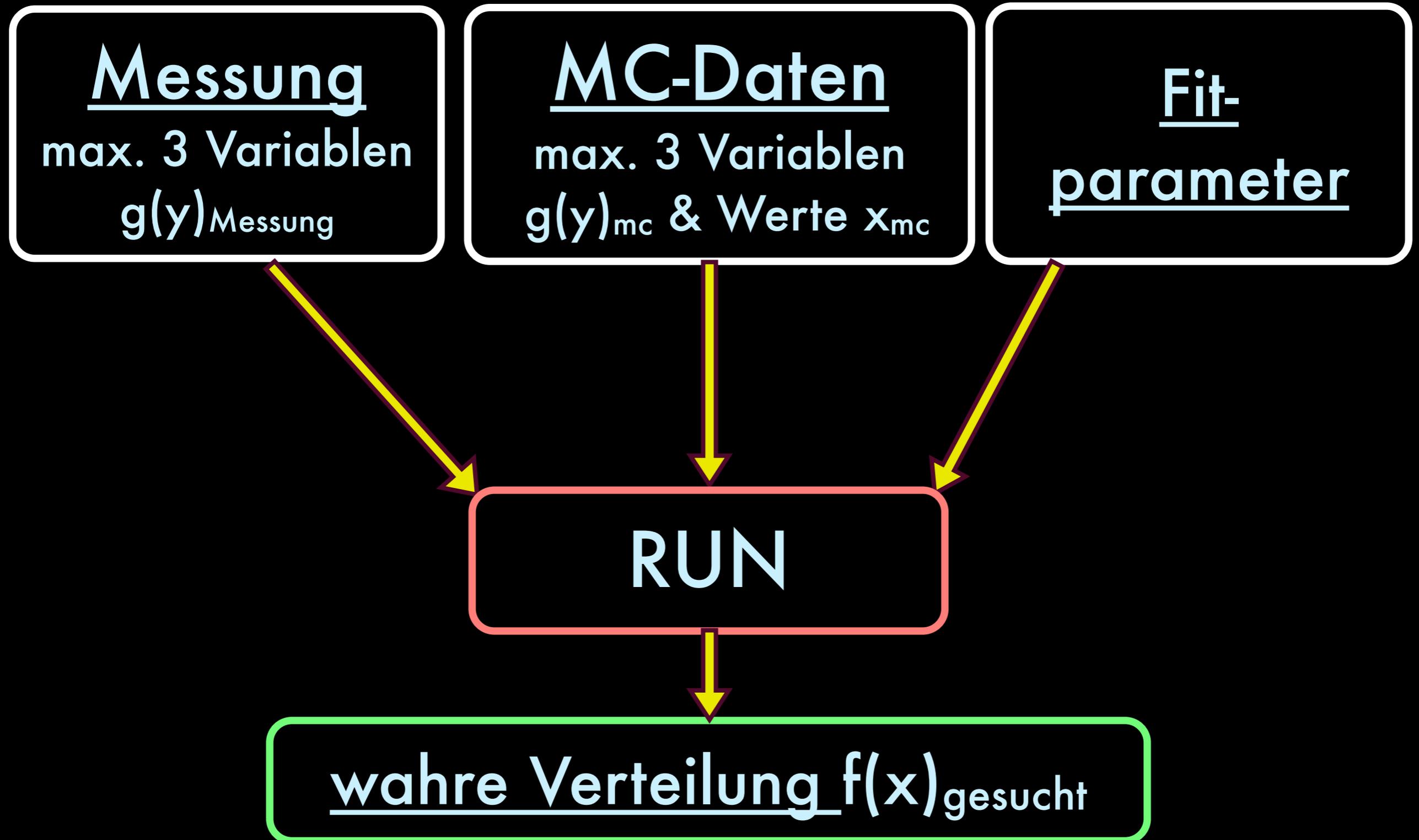
# Entfaltungsprogramm RUN

- Regularized UNfolding (V. Blobel), FORTRAN
- Rekonstruktion der wahren Verteilung:
  - fitten der MC-Daten an die echten Daten
  - gleichzeitig optimieren der Krümmung der gesuchten Verteilungsfunktion (Regularisierung)

$$r(x) = \int (f''(x))^2 dx$$

neg. Log-Likelihood + Krümmung = minimal

# Entfaltungsprogramm RUN



# Entfaltungsprogramm RUN

- liefert sehr gute Ergebnisse, wird u.a. in Dortmund zur Energie-Rekonstruktion bei IceCube und Magic benutzt
- Programm ist leider alt, umständlich zu bedienen (mit ROOT)
- nur 3 gemessene Variablen gleichzeitig einsetzbar
- Auswahl an Fitparametern muss per Hand vorgenommen werden

# RUN-Verbesserungen

- Programm ins C++ übersetzen (ROOT)
- Algorithmus genauer studieren
- Zahl der Variablen für Entfaltung erhöhen
- Optimierung der Auswahl der Fitparameter einbauen
  - ☑ ein Optimierungsalgorithmus ist bereits vorhanden (V. Curtef)

# Optimierungsalgorithmus

- Fitparameter:

- Anzahl der Bins der Lösung
- Grenzen, in denen entfaltet wird
- Zahl der Freiheitsgrade (Einfluss der Regularisierung)
- usw.

beeinflussen die Lösung

# Optimierungsalgorithmus

- Alle Kombinationen durchprobieren kostet zu viel Zeit ( $> 2$  Jahre)
  - ➔ Optimierungsalgorithmus nötig für eine effektivere Suche

# Optimierungsalgorithmus

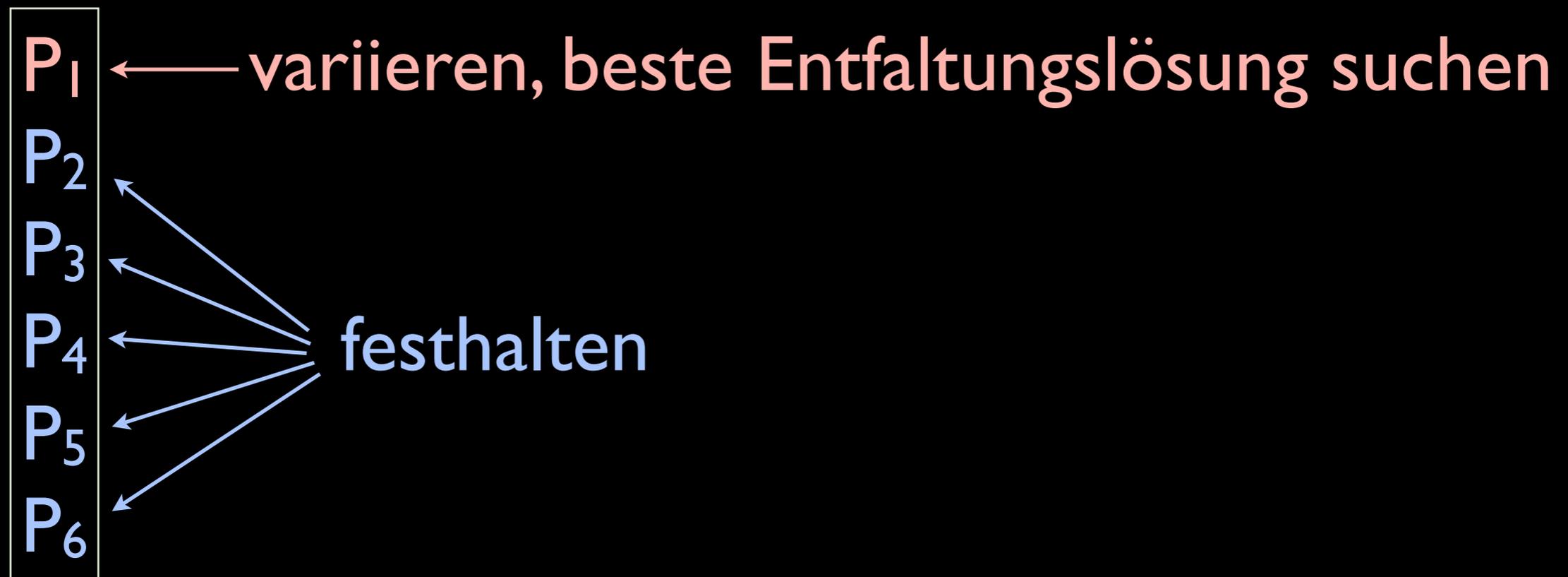
- Alle Kombinationen durchprobieren kostet zu viel Zeit ( $> 2$  Jahre)
  - ➔ Optimierungsalgorithmus nötig für eine effektivere Suche

Betrachte 6 verschiedene Fitparameter

$P_1, P_2, P_3, P_4, P_5, P_6$

# Optimierungsalgorithmus

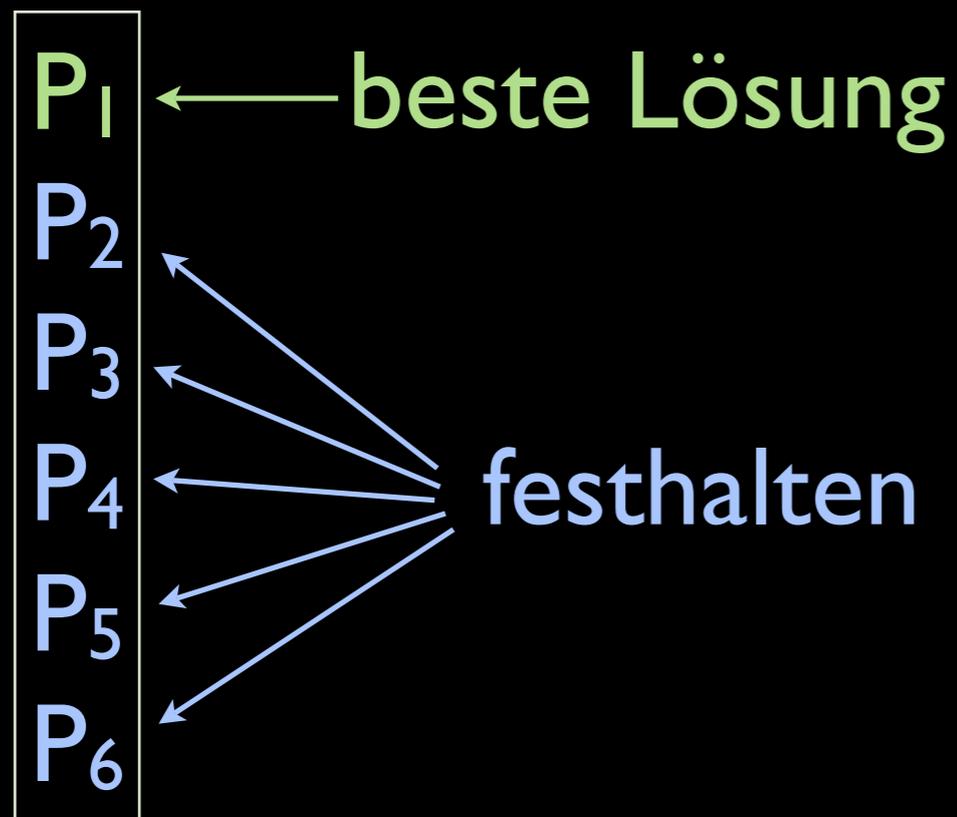
Betrachte 6 verschiedene Fitparameter



Startwerte können willkürlich gewählt werden

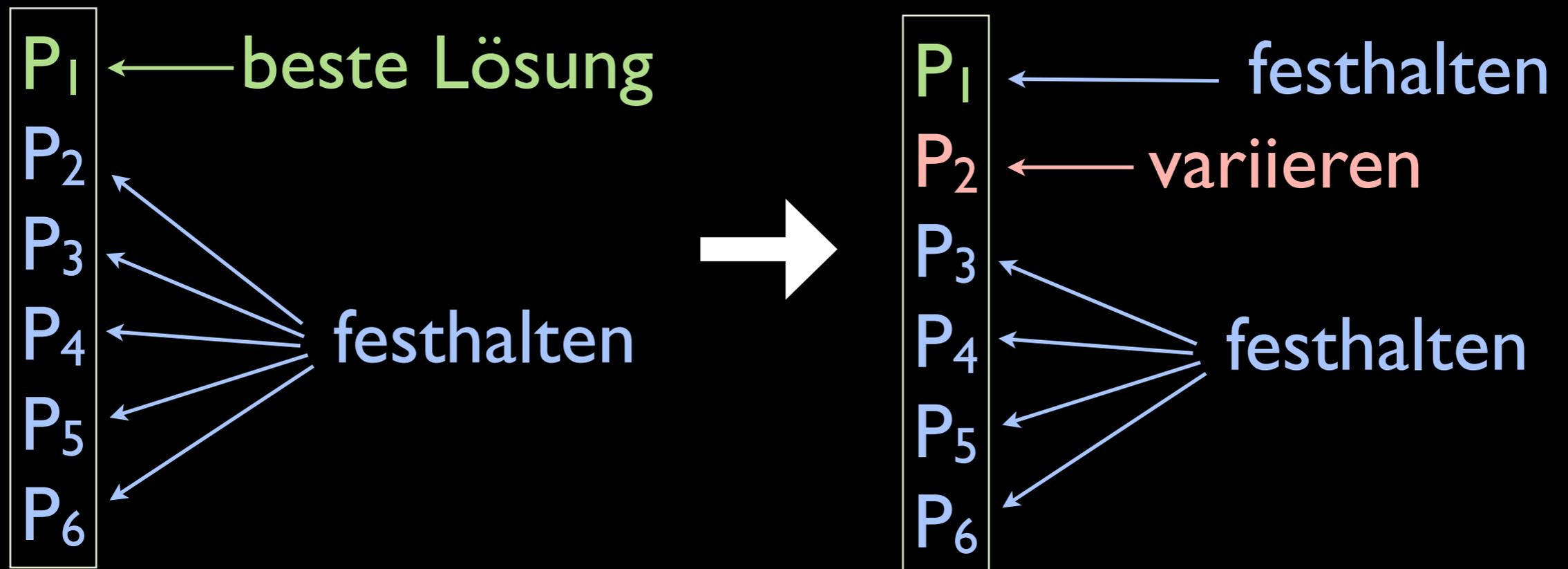
# Optimierungsalgorithmus

Betrachte 6 verschiedene Fitparameter



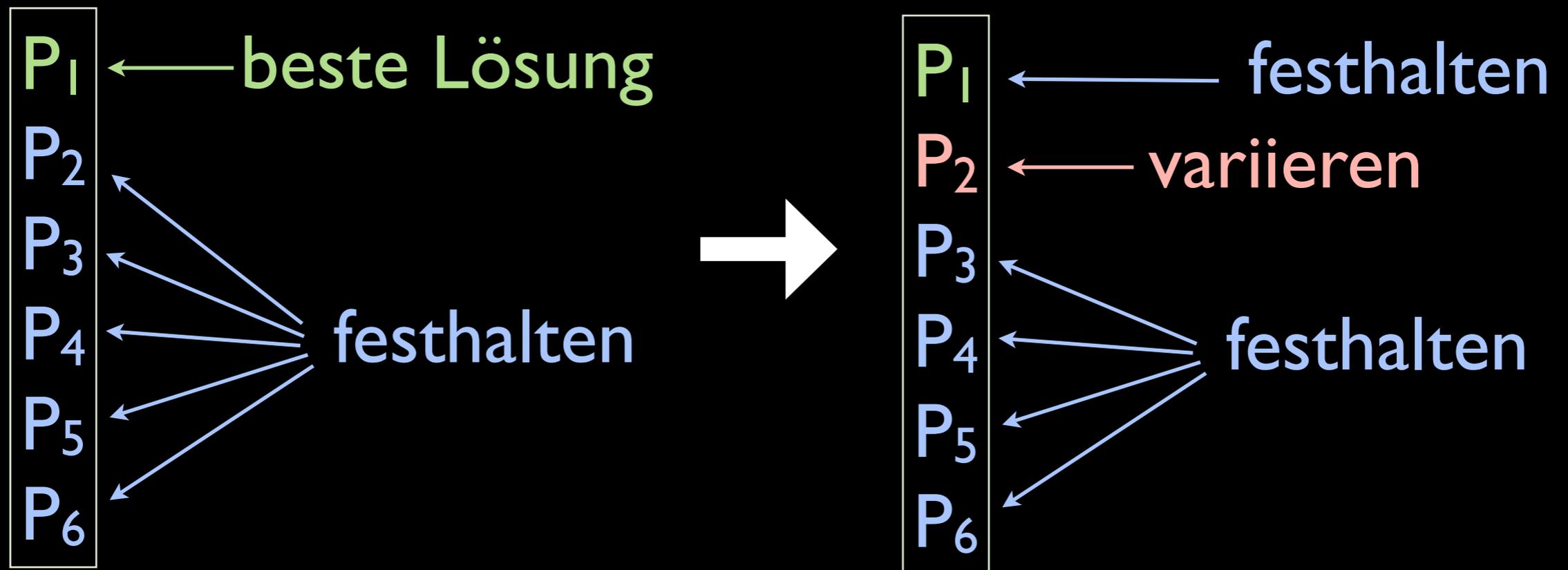
# Optimierungsalgorithmus

Betrachte 6 verschiedene Fitparameter



# Optimierungsalgorithmus

Betrachte 6 verschiedene Fitparameter



usw. bis alle Parameter 1x variiert wurden  $P_1, P_2, P_3, P_4, P_5, P_6$

Mit den neuen Parametern wiederholen

➔ Lösung konvergiert gegen lokales Optimum

# Zusammenfassung + Ausblick

- Regularisierte Entfaltung ist kompliziert aber gut geeignet für Astroteilchenphysik
- RUN: gutes Entfaltungsprogramm, soll an moderne Anforderungen angepasst werden
- RUN wird momentan ins C++ übersetzt und nach Bedürfnissen modifiziert (IceCube, Magic)
- Ein Optimierungsalgorithmus ist entwickelt und wird benutzt (Magic)