# EFFICIENCY ENHANCEMENT OF BALLOONING STABILITY CALCULATIONS FOR QOS EQUILIBRIA VIA RICHARDSON'S EXTRAPOLATION

R. Sánchez

**Universidad Carlos III**, Madrid, SPAIN

S.P. Hirshman and J.C.Whitson

**Oak Ridge National Laboratory**, Oak Ridge, Tennessee, USA

A.S. Ware

**University of Montana**, Missoula, Montana, USA

US-Japan Workshop on Stellarator/Helical System Concept Improvement

Oak Ridge, Tennessee, USA, September 23-24, 1999

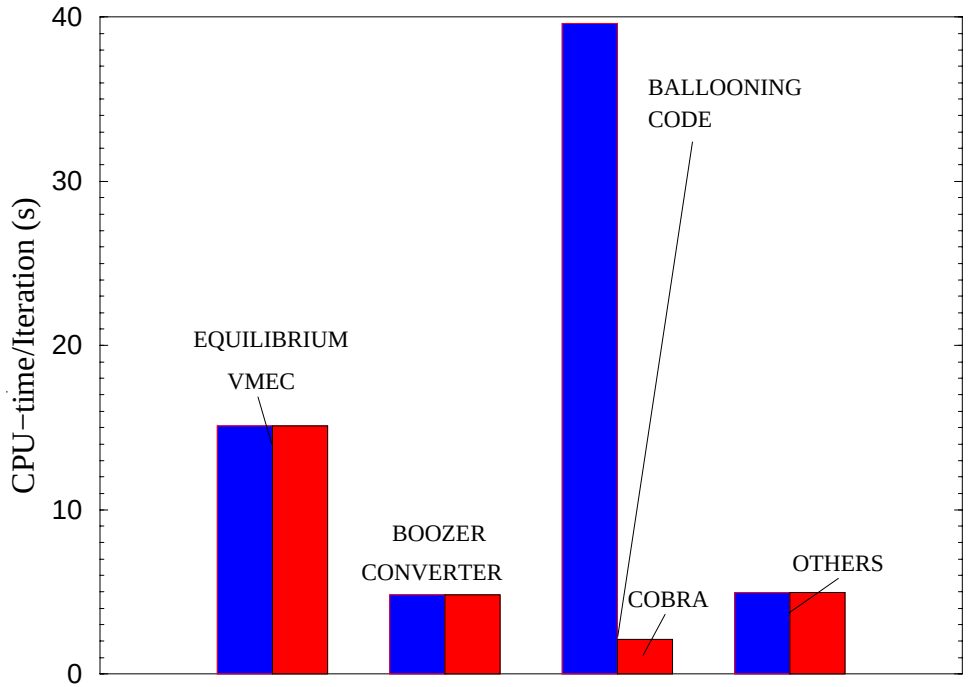
# OUTLINE

- Motivation $\Rightarrow$ Need for faster ballooning codes
- Revisiting the ballooning equation $\Rightarrow$ identify keys for speeding up ballooning codes.
- The COBRA code $\Rightarrow$ more efficient algorithm composed of:
  1. Matrix algorithm
  2. Variational refinement
  3. Richardson's extrapolation
- Comparison of COBRA performance with available codes
- Conclusions

# MOTIVATION

- QOS transport-optimized configurations have very small toroidal currents flowing in the plasma (of the order of a few tens of kA), so ideal ballooning modes set the critical $\beta$.
- Inclusion of ballooning stability targets in transport-optimization procedure reveals itself very inefficient for most available ballooning codes.
- Post-optimization analysis of transport-optimized QOS configurations reveals that the optimizer tends to go to parameter regions with poor ballooning stability properties ($\beta_c \sim 2\%$).
- Need for more efficient, faster and more accurate ballooning algorithms for 3-D magnetic equilibria is apparent.

# CPU-TIME DISTRIBUTION FOR ONE ITERATION IN OPTIMIZATION LOOP




Typical QOS Equilibrium Parameters:

31 surfaces, 886 Boozer modes, 16 ballooning growth rates on 8 surfaces.

# THE IDEAL BALLOONING EQUATION REVISITED

- Eigenvalue problem: $2^{\mathrm{nd}}$-order ordinary linear differential equation with non-constant coefficients.

$$[L_0(y) + \lambda R(y)]\, F = 0 \tag{1}$$

with

$$L_0(y) \equiv \frac{\mathrm{d}}{\mathrm{d}y}\left[P(y)\frac{\mathrm{d}}{\mathrm{d}y}\right] + Q(y) \tag{2}$$

and $P(y)$ and $R(y)$ positive.

- The ballooning growth rate is given by $\gamma^2 = -\lambda$.
- It has to be solved along the magnetic field line $\Rightarrow$ formulated more easily using straight-line magnetic coordinates.

# WHICH IS THE COMPUTATIONAL BOTTLENECK OF BALLOONING CODES FOR 3-D CONFIGURATIONS?

- Representation of 3-D equilibria is done using inverse Fourier series. In Boozer coordinates $(s, \theta, \xi)$, up to 1000 modes are required.

$$
\begin{aligned}
E(s,\theta,\xi) &= \sum_{m,n} E_{mn}(s)\cos(m\theta - nM\xi) \\
O(s,\theta,\xi) &= \sum_{m,n} O_{mn}(s)\sin(m\theta - nM\xi)
\end{aligned}
\tag{3}
$$

- Most time spent in Fourier inverting of the series at each point of the grid along the field line $\Rightarrow$ Total computational time scales linearly with number of points when number of modes exceeds 100.

# KEY FOR SPEEDING UP BALLOONING CODES?

- 2-D CONFIGURATIONS: More efficient algorithm to solve for the eigenvalue

  $\Rightarrow$ Combine matrix and variational schemes!!

- 3-D CONFIGURATIONS: Use the coarsest grid along the line possible to get eigenvalue within required accuracy

  $\Rightarrow$ Use Richardson's deferred approach to the limit!!

# THE COBRA CODE

(COde for Ballooning Rapid Analysis)

MATRIX ALGORITHM + VARIATIONAL REFINEMENT

+

RICHARDSON'S EXTRAPOLATION

# MATRIX ALGORITHM

- Ballooning equation is discretized and cast into matrix form to second order accuracy:

$$A \cdot \vec{F}^{\mathrm{mat}} = \lambda^{\mathrm{mat}} \vec{F}^{\mathrm{mat}}, \qquad \vec{F}^{\mathrm{mat}} = (F_1, ..., F_j, .., F_N) \tag{4}$$

- Eigenvalue and eigenfunction obtained via standard techniques for non-symmetric tri-diagonal matrices:

$$\begin{aligned} \lambda^{\mathrm{mat}} &= \lambda^0 + \alpha h^2 + O(h^3) \\ F^{\mathrm{mat}} &= F^0 + h^2 \delta F + O(h^3) \end{aligned} \tag{5}$$

- Important to notice the eigenfunction is $2^{\mathrm{nd}}$-order accurate!!!

# VARIATIONAL REFINEMENT

- Rewrite ballooning equation as variational principle $\Rightarrow$ local minima of functional $H$:

$$H(F^*, F) = -\frac{< F, L_0 F >}{< F, RF >}; \quad < F, G > \equiv \int_{-\infty}^{\infty} F^*(y) G(y) dy \quad (6)$$

  in space of integrable functions yield eigenfunctions, and the functional value the eigenvalues.

- In the neighbourhood of any minimum, first-order variation of the functional vanishes:

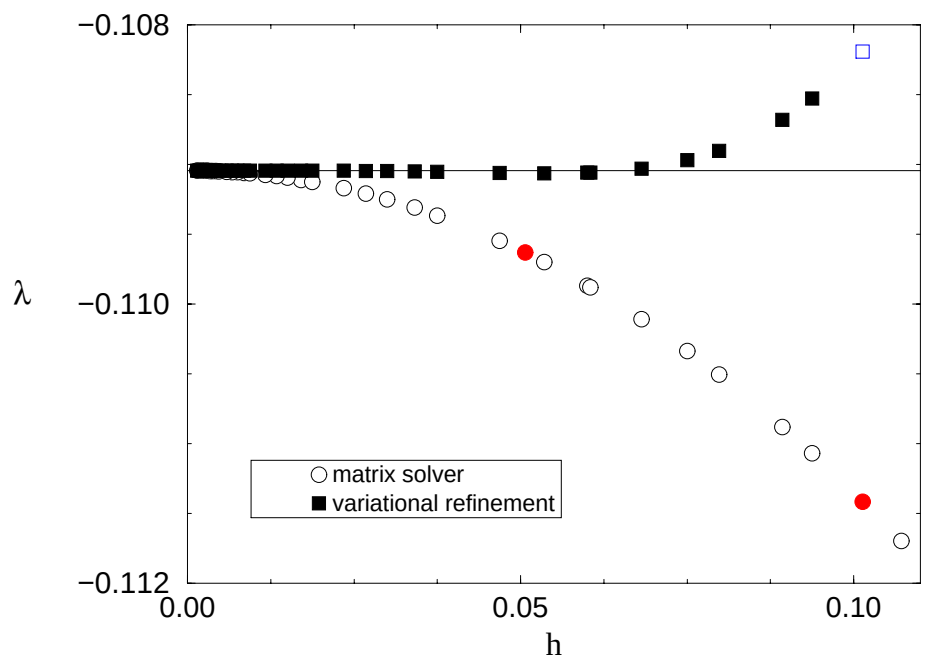
$$H(F_0^* + \delta F^*, F_0 + \delta F) = H(F_0^*, F_0) + \delta^2 H(\delta F^*, \delta F) + .... \quad (7)$$

- Eigenfunction $F^{\mathrm{mat}}$ from matrix method is $2^{\mathrm{nd}}$-order accurate
  $\Rightarrow$ a $4^{\mathrm{th}}$-order accurate estimation of eigenvalue is obtained when inserting it into the functional $H$:

$$\lambda^{\mathrm{var}} = \lambda_0 - h^4 \left[ \frac{\langle \delta F, L_0(y) - \lambda_0 R(y) \delta F \rangle}{\langle F, R(y) F \rangle} \right] + O(h^5) \qquad (8)$$

# WHY IS IT MORE EFFICIENT?

- Highest order reduces up to five times number of grid points for a prescribed accuracy ⇒ 5 times faster algorithm!!



# RICHARDSON'S DEFERRED APPROACH TO THE LIMIT

- WHY DO WE NEED IT?: The previous algorithm is five times faster than standard second-order shooting methods, but provides with NO way to choose the right value for the grid step size.

- DESCRIPTION OF THE METHOD:

  1. Assume eigenvalue can be somehow computed to $n^{\text{th}}$-order occuracy:

$$\lambda = \lambda_0 + \alpha h^n + O(h^{n+1}) \tag{9}$$

2. Compute eigenvalue for first few elements of a decreasing series for the grid step size:

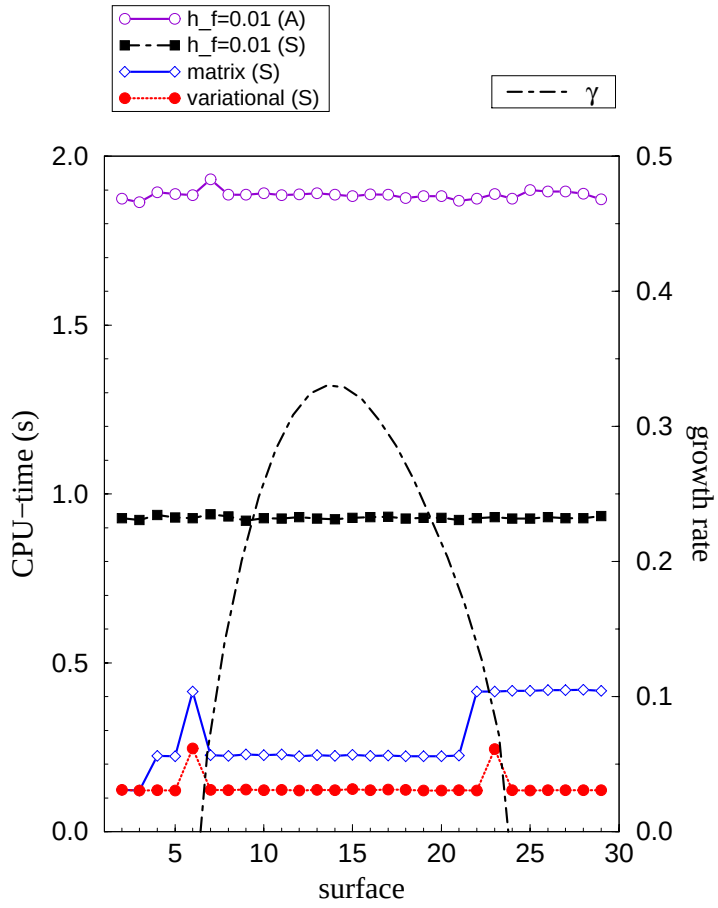
$$h_k = \frac{h_{k-1}}{2} \tag{10}$$

and extrapolate to $h \to 0$ to get $\lambda_0$ using previous power law.

3. Estimate extrapolation error and if not below prescribed tolerace, iterate including next step size in the series.

- Choosing very large values for $h_0$, the coarsest mesh is found at run-time $\Rightarrow$ 10 times faster
- The higher value of $n$ the faster the extrapolation error diminishes $\Rightarrow$ extra advantage from higher order variational refinement (n=4).

# EFFICIENCY ENHANCEMENT VIA RICHARDSON'S EXTRAPOLATION?


h_f=0.01 (A)
h_f=0.01 (S)
matrix (S)
variational (S)
γ
CPU−time (s)
growth rate
surface
2.0
1.5
1.0
0.5
0.0
0.5
0.4
0.3
0.2
0.1
0.0
5
10
15
20
25
30

# OTHER COBRA FEATURES

- Uses the $y \to -y$ symmetry along the magnetic field line existent at certain initial points in configurations with stellarator symmetry to halve the number of grid points $\Rightarrow$ 2 x 10 times faster
- Incorporates criteria to decide at run-time the required length along the magnetic field line.
- Ability for solve for any eigenvalue in the discrete part of the spectrum without losing efficiency.
- Can equally explore stable part of spectrum $\Rightarrow$ Application to high-n Alfvèn modes appearing in continuous spectrum gaps!

# COBRA IMPROVEMENTS RELATIVE TO AVAILABLE BALLOONING CODES

- Most available ideal ballooning codes use either shooting methods (with fixed or adaptative grid step size) or optimization algorithms to minimize the function $H$ within a family of test functions.
- Improvement relative to fixed step size algorithms is clear:

  $\Rightarrow$ up to 20 times faster!!

  $\Rightarrow$ ensures eigenvalue convergence at run time!!
- Adaptative grid schemes ensure convergence but in practice are even slower, since more and more grid points need to be computed when refining the mesh!
- Variational methods yield a lower limit for the growth rate

  $\Rightarrow$ Necessary but not suficient condition for stability!!

# CONCLUSIONS

- Use of Richardson's extrapolation coupled to a more efficient algorithm has allowed to increase up to 10-20 times the efficiency of ballooning stability calculations for 3-D configurations.
- A fast ideal ballooning code, COBRA, has been developed including all these capabilities.
- By using COBRA, ballooning targets can now be efficiently included in QOS optimization suite.