Domain decomposition spectral method for solving two-point boundary value problems

Manfred Trummer
Tracy Obison
Simon Fraser University
Canada

Go20 - Gozo, Malta

May 23, 2025

Singular perturbation problem

$$\epsilon z''(x) + p(x)z'(x) + q(x)z(x) = f(x),$$

$$x \in (-1, 1), \quad z(-1) = \alpha, \quad z(1) = \beta,$$

- ε is a small parameter

- Leads to thin boundary or interior layers

- Typical width of layer is O(ε)

Layers and corners

Resolving boundary layers

- Use graded meshes, with many mesh points near the boundaries, e.g. Shishkin mesh

- Use coordinate transformations , e.g. on [-1,1]
$$y = \sin\left(\frac{\pi}{2}x\right)$$

- Neither approach is effective for interior layers

Spectral Methods

- High order methods

- Exponential convergence

- Based on global interpolation/approximation

- Sensitive

Chebyshev Differentiation

- Given a function f, approximate f'

- Interpolate f at the N+1 Chebyshev points $x_k = cos(k\pi/N)$

  with a polynomial p of degree N (or less)

- Differentiate p $\rightarrow$ p' exactly

- Evaluate p' at the Chebyshev points $\rightarrow$ approximates f'

- All operations are linear!

Chebyshev Differentiation

$$x_k = \cos(k\pi/N)$$

Differentiation Matrix:

$$D_{kj} = \frac{c_j}{c_k} \frac{1}{x_k - x_j}, \qquad k \neq j,$$

$$D_{kk} = -\frac{1}{2} \frac{x_k}{1 - x_k^2}, \qquad k \neq 0, N,$$

$$D_{00} = -D_{NN} = \frac{2N^2 + 1}{6},$$

where $c_0 = \frac{1}{2}, \quad c_N = \frac{1}{2}(-1)^N, \quad c_k = (-1)^k.$
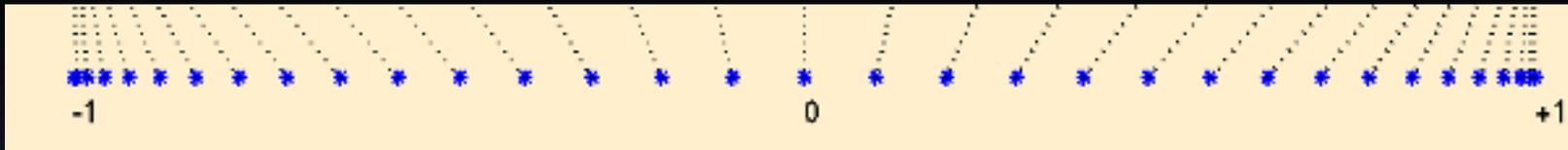
NST: Negative sum trick

Spectral Differentiation is exact for constant functions, hence

$$D_{kk} = -\sum_{j=0,j\neq k}^{N} D_{kj}$$

Arrange order of summation to sum the smaller elements first - requires sorting

Spectral methods and layers

- Use Chebychev points as collocation points

- Tends to work well for **boundary** layers

- To numerically resolve layers at least one collocation point should be in the layer

Solving a Two-Point BVP

$$\epsilon z''(x) + p(x)z'(x) + q(x)z(x) = f(x),$$

$$x \in (-1, 1), \quad z(-1) = \alpha, \quad z(1) = \beta,$$

$$A = \epsilon D^{(2)} + PD + Q;$$
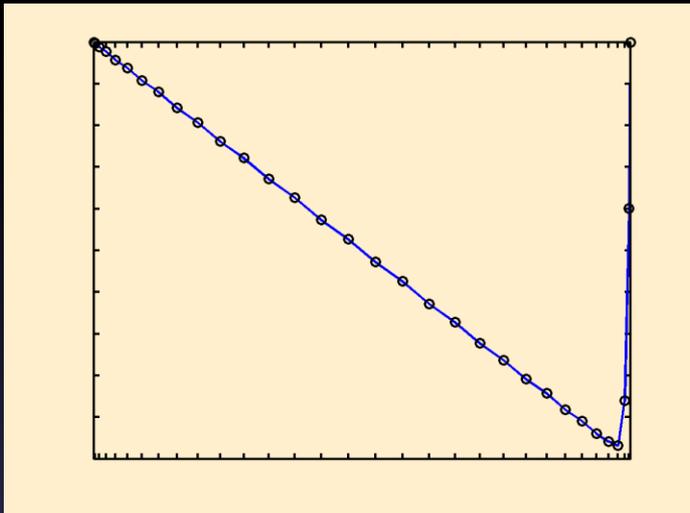
$$b = f - \alpha A(:, N) - \beta A(:, 1).$$

$$D^{(2)} = D^2$$

Including BCs

- Condition number κ(A) is "large" – $O(N^4)$

- The first and last row/column of A are "removed" and "replaced" by boundary conditions

- The first and last column of the differentiation matrix reappear in the right-hand side

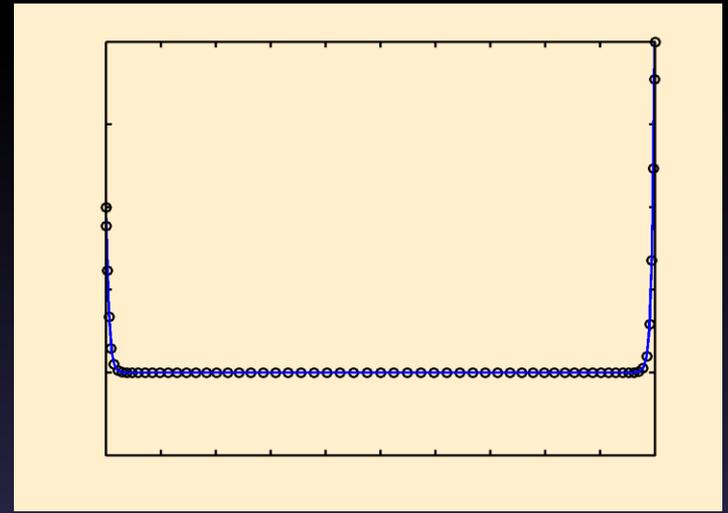- Zero boundary condition: little amplification of errors

Model Problems

Model 1

Model 2

$$\epsilon u'' - u' = \frac{1}{2}$$

$$\epsilon u'' - xu' - u = f$$

$$u(-1) = u(1) = 0$$

$$u(-1) = 1; \quad u(1) = 2$$

Accuracy for ε = 0.01

N	Model 1	Model 2
32	6.8E-03	9.3E-03
64	1.4E-09	1.4E-09
81	2.0E-14	2.9E-14
128	2.4E-14	4.0E-14
200	1.8E-13	1.4E-13

So, no problem for this ε - need to look at smaller ε !

Model 1. Accuracy for smaller ε

Mean-Square Error: $||z_{num} - z_{exact}||_2 / \sqrt{N}$

ε = 10⁻⁴

N	MS-Error
128	2.55E-01
256	1.18E-02
512	3.05E-07
1024	3.44E-14
2048	4.10E-14

ε = 10⁻⁶

N	MS-Error
128	4.28E+01
256	1.04E+01
512	2.38E+00
1024	4.57E-01
2048	4.86E-02

Breaks down for ε = 10⁻⁸

Domain Decomposition

Break the interval into two parts

$$\gamma = 1 - \sqrt{\epsilon}$$
Intervals $[-1 \quad \gamma]$ and $[\gamma \quad 1]$

Interface condition: $z(\gamma) = c$ "c is unknown"

Solve on each interval obtaining $z_L(x)$ and $z_R(x)$

Determine c by solving $z'_L(\gamma) = z'_R(\gamma)$

Use the **secant method** to find the solution for "c" – FAST!

Results – Model 1

$\varepsilon = 10^{-4}$

Domain Decomposition

total N	MS-Error
128	4.62E-14
256	6.70E-14
512	4.55E-14

Classical

N	MS-Error
128	2.55E-01
256	1.18E-02
512	3.05E-07
1024	3.44E-14
2048	4.10E-14

Results – Model 1

$\varepsilon = 10^{-6}$

Domain Decomposition

total N	MS-Error
128	1.38E+00
256	7.38E-07
512	2.93E-12
1024	2.72E-12

Classical

N	MS-Error
128	4.28E+01
256	1.04E+01
512	2.38E+00
1024	4.57E-01
2048	4.86E-02

Results – Model 1

$\varepsilon = 10^{-8}$

Domain Decomposition

Classical

total N	MS-Error
512	5.32E-01
1024	1.85E-10
2048	7.69E-11
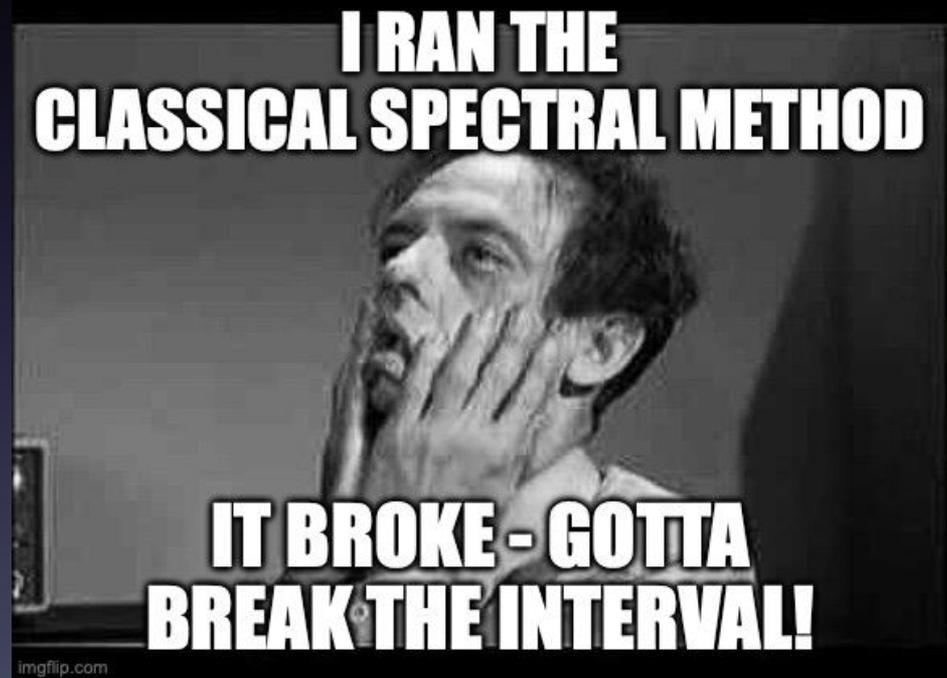
Cost

- Secant method. Even with two fairly bad initial guesses, typically only 2 iterations are required.

- Cost of solving two problems of size N/2 is $2*C*N^3/8$ compared to $C*N^3$ for problem of size N

- So the domain decomposition method with a total of N points has basically the same cost as the classical spectral methods with N points.

Model 2: Accuracy for smaller ε

Mean-Square Error: $||z_{num} - z_{exact}||_2 / \sqrt{N}$

ε = 10^{-4}

N	MS-Error
128	1.08E+00
256	3.22E-02
512	6.66E-07
1024	6.77E-14
2048	1.15E-13

ε = 10^{-6}

N	MS-Error
512	2.54E+01
1024	6.08E+00
2048	4.14E-01
4096	2.57E-04

Breaks down for ε = 10^{-8}

Domain Decomposition

Break the interval into THREE parts
Intervals $\quad [-1 \ \ \gamma_L], \ \ [\gamma_L \ \ \gamma_R], \ \ [\gamma_R \ \ 1]$

$$\gamma_L = -1 + \sqrt{\epsilon}, \ \ \gamma_R = 1 - \sqrt{\epsilon}$$

Interface condition: $\quad z(\gamma_L) = c_L, \quad z(\gamma_R) = c_R$

Find $\vec{c} = \begin{bmatrix} c_L \\ c_R \end{bmatrix}$ by matching derivatives of the

corresponding solutions for each subinterval at the
interface points

Match derivatives

- System of nonlinear equations $F(\vec{c}) = 0$

- Solve via secant = Broyden's method

- Quasi-Newton, with the approximation to the Jacobian updated with new secant information

Broyden

Broyden $\quad F_k = F(c_k)$

$d_n = c_n - c_{n-1}.$ $\quad$ Want $\quad J_n d_n = F_n - F_{n-1}$

Update Jacobian:

$J_n = J_{n-1} + \frac{1}{||d_n||^2} \left(F_n - F_{n-1} - J_{n-1} d_n \right) d_n^T$

$c_{n+1} = c_n + \mu J_n^{-1} F_n$

note: $\mu = 1$ for full Newton step

Start with $\quad J_0 = I \quad$ identity

Results – Model 2

$\varepsilon = 10^{-4}$

Domain Decomposition

total N	MS-Error
66	1.25E-01
96	2.05E-04
126	1.10E-07
192	8.44E-14

Classical

N	MS-Error
128	1.08E+00
256	3.22E-02
512	6.66E-07
1024	6.77E-14
2048	1.15E-13

Results – Model 2

$\varepsilon = 10^{-6}$

Domain Decomposition

total N	MS-Error
192	6.37E+00
384	2.66E-06
600	4.32E-12
768	5.34E-12
1500	3.85E-12

Classical

N	MS-Error
512	2.54E+01
1024	6.08E+00
2048	4.14E-01
4096	2.57E-04

Results – Model 2

$\varepsilon = 10^{-8}$

Domain Decomposition

total N	MS-Error
768	2.99E+00
1002	1.27E-02
1500	4.34E-09
2004	2.39E-10

Classical
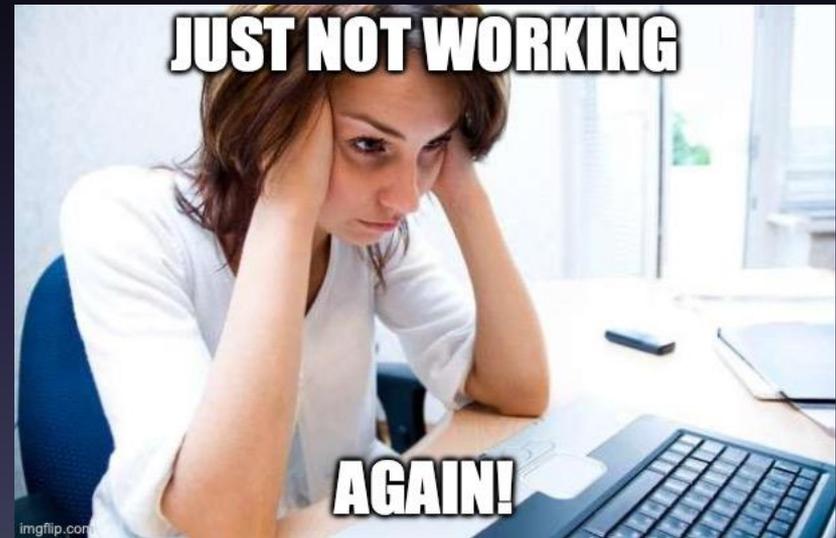
Back to Model 1 with one layer

- Make 3 intervals , $\varepsilon=10^{-6}$

- Interface points at 0.999 and 0.9999

Results – Model 1

$\varepsilon = 10^{-8}$

3 Intervals

Interface points at 0.9999 and 0.99999

total N	MS-Error
126	2.91E-07
270	2.17E-12
504	3.22E-12

2 Intervals

total N	MS-Error
128	1.38E+00
256	7.38E-07
512	2.93E-12
1024	2.72E-12

m>3 subintervals

- Easily generalizes to m>3 subintervals

- Derivative matching is now a nonlinear system of (m-1) equations in (m-1) unknowns.

- Cost. With a total of N points, solving the m small problems will incur a cost of $C*N^3/m^2$ compared to $C*N^3$

- Broyden typically converges with 4-6 evaluations of F → Efficiency increases with m
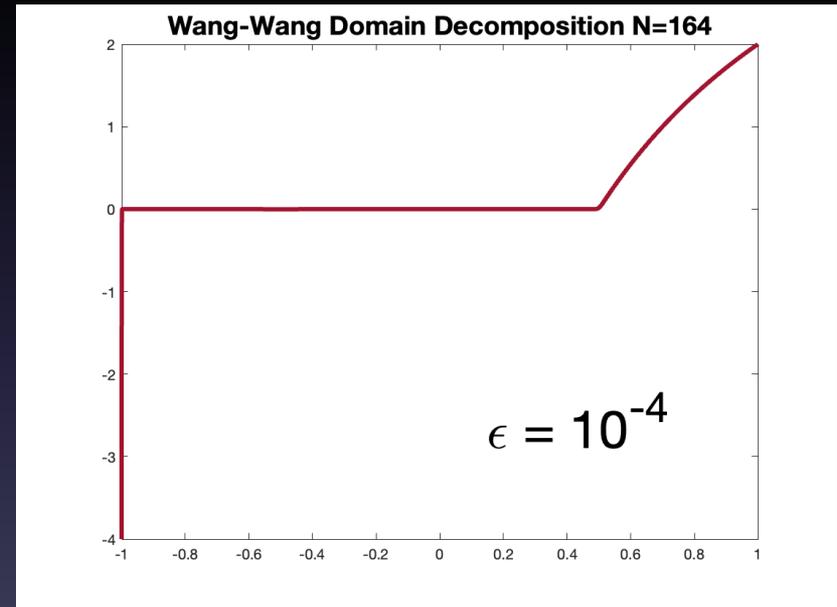
Results – Model 2

$\varepsilon = 10^{-10}$

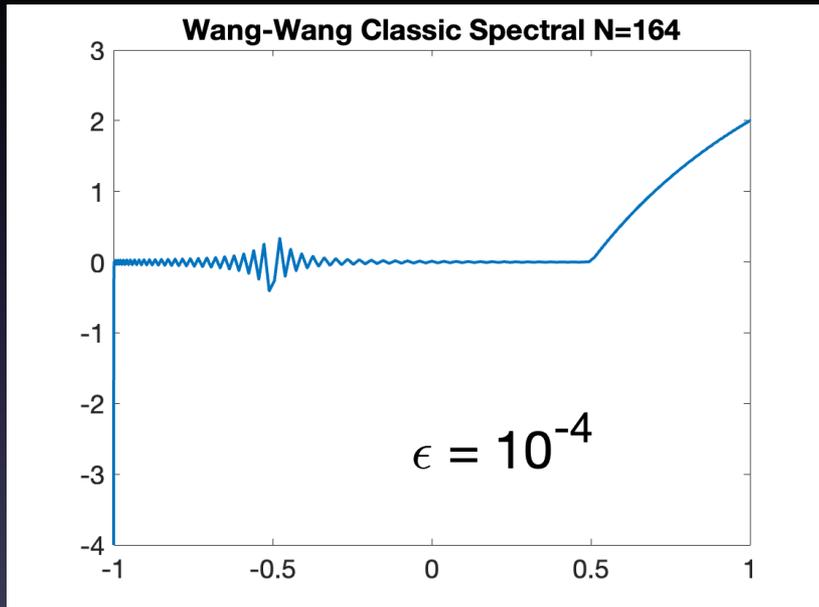m=5 Intevals with Interface points at

$$\pm(1 - 5 * 10^{-8}) \quad \text{and} \quad \pm(1 - 10^{-5})$$

total N	MS-Error
400	1.62E+00
600	1.99E-07
620	2.88E-08
640	1.36E-08
1000	2.76E-08

Layer at left, corner point at 1/2

$$\epsilon z'' + (x+1)\left[(x+\tfrac{1}{2})(x-\tfrac{1}{2})z' - z\right] = 0,$$
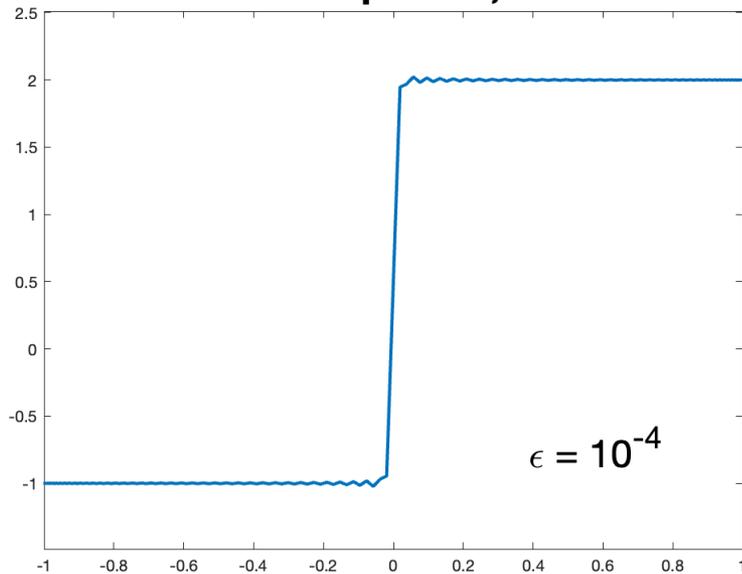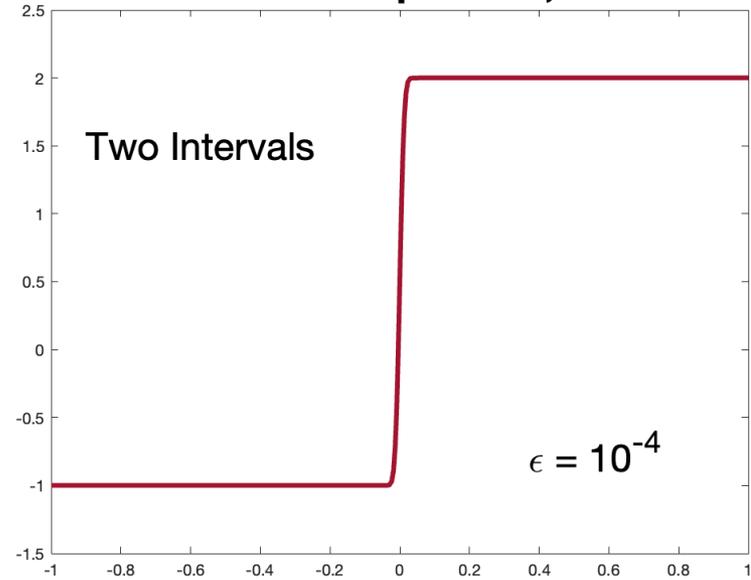$$z(-1) = -4, \quad z(1) = 2.$$

5 subintervals

Interior Layer

$$\epsilon z'' + xz' = 0,$$
$$z(-1) = -1, \quad z(1) = 2.$$

Classical Spectral, N=164

$\epsilon = 10^{-4}$

Domain decomposition, N=164

Two Intervals

$\epsilon = 10^{-4}$

Summary

- Domain decomposition with derivative matching is efficient and accurate

- Secant and Broyden converge nicely (why?)

- Can deal with interior layers

- Open to adaptivity

To Do

- More testing

- Investigate convergence properties of Secant and Broyden's method

- Choice of interface points

- Adaptivity – move interface points, different "N" for different subintervals