

Linear Analysis and Verification Suite
for Edge Turbulence:
Final Report for the 2DX Project

J. R. Myra, D. A. Baver

Lodestar Research Corp., 2400 Central Ave. P-5, Boulder, Colorado 80301

M. V. Umansky

Lawrence Livermore National Laboratory, Livermore, CA 94550

July 2011

unpublished

DOE/ER/84718-4

LRC-144

LODESTAR RESEARCH CORPORATION
2400 Central Avenue
Boulder, Colorado 80301

Table of Contents

I. Executive summary	3
II. Description of 2DX functionality and usage	4
A. Equation language	4
B. Divertor geometry	6
III. Physics models	6
A. Advanced fluid models	6
B. Kinetic extensions	7
IV. Benchmark-verification tests and debugging applications	8
A. Development of benchmark tests	8
B. Application of 2DX as a debugging aid	10
V. Physics applications	11
VI. Conclusions	14
Acknowledgments	14
References	15

Appendix A: Six-field model equations

Appendix B: Resistive ballooning mode benchmark

Appendix C: Resistive drift mode benchmark (ES and EM)

Appendix D: Slab ion temperature gradient (ITG) mode benchmark

Appendix E: Geodesic acoustic mode (GAM) benchmark

Appendix F: Kink mode benchmark

Appendix G: Parallel Kelvin-Helmholtz mode benchmark

Appendix H: Toroidal ion temperature gradient mode (ITG) benchmark

Appendix I: Edge localized mode (ELM) benchmark

Appendix J: Kinetic resistive ballooning mode benchmark

Appendix K: D.A. Baver et al., *Comp. Phys. Comm.* **182**, 1610, (2011).

I. Executive summary

The edge and scrape-off-layer region of a tokamak plasma is subject to well known drift, resistive and ideal instabilities that are driven by various curvature, gradient, and sheath related mechanisms. Sometimes a single mode appears to dominate the edge plasma;^{1,2} often the edge is turbulent.^{3,4} In either case, it is useful to have computational tools that can analyze the linear eigenmode structure, predict quantitative trends in growth rates and elucidate the underlying drive mechanisms. Furthermore, measurement of the linear growth rates of unstable modes emerging from a known, established equilibrium configuration provides one of the few quantitative ways of rigorously benchmarking large-scale plasma turbulence codes with each other and with a universal standard. These needs have motivated development of a suite of codes, collectively referred to as 2DX, that can describe linearized, nonlocal (i.e. separatrix-spanning) modes in axisymmetric (realistic divertor), toroidal geometry.

The 2DX code⁵ employs a generalized eigenvalue technique, and as such possesses some important inherent advantages over computations in the time domain. These include the ability to use realistic collisionality parameters, freedom from time-step limitations, and the ability to discard unphysical numerical modes by post-processing. These advantages usually result in run times that can be one or more orders of magnitude faster than time-stepping techniques. Moreover, a novel method of coding physics models within 2DX is extremely flexible and less prone to coding errors than traditional programming approaches. This makes 2DX particularly attractive for developing benchmarking standards, as well as for exploring and comparing the consequences of different physics models. The 2DX suite includes a number of tools for setting up problems, verifying inputs, and post-processing outputs, as described in Sec. II of this report.

A wide range of physics models has been implemented in 2DX, from simple specialized one- or two-field models, to advanced multi-field models, which are capable of including some kinetic effects. The implemented models are described in Sec. III, and include a complete linearization of the six-field fluid model employed in BOUT.^{6,7} Every important edge and scrape-off-layer mode discussed in the contemporary literature is embedded in this model.

In Sec. IV, a large set of benchmark comparisons for the models are tabulated. The comparisons are with both semi-analytical results (for simplified geometry cases)

and with the BOUT^{6,7} and BOUT++⁸ edge turbulence codes (in full divertor and X-point geometry). These tests serve to (i) verify the coding of the physics models within 2DX, (ii) verify both 2DX and BOUT in cases where they have been compared, and (iii) provide solid results which other codes can employ as benchmark targets. Self-contained descriptions of benchmark tests have been given to the Edge Coordination Committee (ECC) for display on the V&V section of their website,⁹ and are also available through the Lodestar web site.¹⁰ In one case, for a model ELM problem, four codes (2DX, ELITE, GATO and BOUT++) were all successfully benchmarked against each other. An example of how the 2DX code has been used as a debugging aid is also given.

Several physics applications of 2DX have already been carried out; these just begin to scratch the surface of possible 2DX applications. In Sec. V we describe a study of parallel disconnection induced by X-point effects (relevant to blob propagation^{11,12}) and an analysis of candidates for the quasi-coherent mode in Alcator C-Mod.² Finally, some future applications and extensions are discussed in Sec. VI.

The 2DX project has succeeded in developing a very versatile and flexible computational tool for application to edge physics problems in magnetic fusion plasmas. The key features which distinguish 2DX are: (i) the use of eigenvalue methods to enable accurate, high resolution studies with modest computation effort, and (ii) a novel, convenient and human-error-minimizing approach for implementing a wide variety of physics models. Additionally, with the completion of the project, there now exists a rather complete set of edge-instability benchmark-verification tests¹³ already passed by 2DX (and available for community use). Thus, the 2DX code provides reliable mode analysis and benchmarking capabilities for edge physics research which, to the best of our knowledge, are unique within the fusion community.

II. Description of 2DX functionality and usage

The 2DX code includes two features which make it particularly useful for benchmarking and physics applications to problems in tokamak edge research: (i) an equation language input capability, and (ii) the ability to handle complex edge geometry.

A. Equation language

The equation parser built into the 2DX eigenvalue code provides a convenient and error-minimizing approach to coding complex high-fidelity physics models. For example, the full equation set for the six-field model, discussed later in this document, includes

more than 50 terms, and is similar to the equation set implemented in the BOUT code after linearization.^{6,7} Coding of these types of complex equation sets is greatly facilitated by use of the equation language.

The language and some of its associated tools are illustrated here with a simple example, the three-field resistive ballooning model, defined by the following equations (See Ref. 5 and Appendix B for a complete description of the model.)

$$\gamma \nabla_{\perp}^2 [\delta\Phi] = \frac{B^2 \partial_{\parallel} [\delta J]}{n_0} + \nabla_{\perp}^4 [\delta\Phi] \mu_{ii} + \frac{2 B C_r [\delta N (T_e + T_i)]}{n_0} \quad (1)$$

$$\gamma \delta N = -\delta v_E \cdot \nabla n_0 \quad (2)$$

$$\gamma \delta J = -\mu \nabla_{\parallel} [\delta\Phi] n_0 - \delta J v_e \quad (3)$$

where $\delta J = -\nabla_{\perp}^2 \delta A$, and $\delta v_E \propto \delta\Phi$.

After setting up the appropriate field variables and input parameters in the 2DX “structure” file the equation is coded in a straightforward manner. For example Eq. (2) is coded as

$$\text{gg}*(1+0j)*N=(-1+0j)*kbrbpx*n0p*PHI \quad (4)$$

where gg is the structure file notation for the eigenvalue γ , complex constants are in parenthesis, $N = \delta N$, and $PHI = \delta\Phi$ are field variables, and $kbrbpx$ and $n0p$ represent a function and an input profile respectively. Functions are coded in a manner similar to the language used by an RPN (stack-based) calculator. The structure file contains standard definitions of differential operators in terms of built-in elementary operators for derivatives along each of the coordinate directions. For example

$$\nabla_{\parallel} = \mathcal{J} \cdot \partial_y \quad (5)$$

$$\partial_{\parallel} = \mathcal{J}B \cdot \partial_y \cdot \frac{1}{B} \quad (6)$$

$$\nabla_{\perp}^2 = -\mathcal{J}B \cdot (-i \partial_x \cdot RB_p + k_{\psi}) \cdot \frac{1}{\mathcal{J}B} \cdot (-i RB_p \cdot \partial_x + k_{\psi}) - k_b^2 \quad (7)$$

where x and y are respectively the radial (flux) and parallel field-line-following coordinate directions.

A valuable feature of the 2DX suite, is that various tools, in both python and Mathematica, are available for editing and viewing the equation language structure files. In fact, Eqs. (1) – (3) and (5) – (7) have been typeset by an “equation viewer” from the

same input file that is actually used by the executable code. This feature is quite valuable for checking the coding of complex equation sets. In general, the structure file can define subsidiary functions involving nested definitions, fractions, etc which are unraveled and displayed in the equation-viewer's output.

Other python, Mathematica and IDL toolsets assist in setting up plasma profile and geometry inputs (see the following section), viewing the output, and diagnosing the contributions of individual terms (e.g. to assess the relative importance of various physical effects).

B. Divertor geometry

Geometry is handled in the 2DX code by "gridfiles" and an associated gridfile generator, which maps physical space and functions thereon to the coordinates. In the present version of 2DX, the domain of the grid is divided into four regions. One of these regions is the edge (the region inside the separatrix). It is bounded by periodic boundary conditions along the field line. A second region is the scrape-off layer. This is the region outside the separatrix but adjacent to the edge. It is subject to sheath boundary conditions at the ends of the field line. The other two regions join together to form the private scrape-off layer. (See Ref. 5 attached as Appendix K for a more complete description of the geometry and code architecture.) Full divertor geometry benchmark tests of 2DX with BOUT are reported in Sec. IV.

III. Physic models

A. Advanced fluid models

The most advanced model routinely employed is a collisional edge fluid model that includes equations for six fields: electrostatic potential, density, parallel mass velocity, electron temperature, ion temperature, and parallel vector potential. The equation set is similar to that employed by other authors^{6,7,14} The electrostatic potential is given in the form of a vorticity equation, while the equation for the parallel vector potential combines Ampere's law and the parallel Ohm's law, and thus provides the physics of the parallel current. The equilibrium can include flows and parallel current. Phase-shift-periodic boundary conditions are imposed on closed flux surfaces and sheath boundary conditions are imposed on the open field lines. This physics model, when solved as an eigenvalue problem for mode growth rates, includes familiar edge modes

such as the drift wave, the resistive ballooning mode, the ion temperature gradient mode, edge peeling and kink modes, sheath-driven modes, perpendicular and parallel Kelvin-Helmholtz modes, and many others. Collisionality, ion diamagnetism, background flows, electromagnetism and many other effects are included in this physics description. The equation set for the full six-field model is given in Appendix A.

Frequently is not necessary, or even desirable, to employ the full six-field model with all of its terms. More insight can often be gained by working with reduced equation sets that contain just the physics of most interest for the given application. Many of these subsidiary models have been benchmarked in Sec. IV.

B. Kinetic extensions

The 2DX code is fundamentally fluid-based, in that it solves generalized eigenvalue problems in two dimensions (i.e. the R-Z plane in axisymmetric toroidal geometry). The 2D generalized eigenvalue structure results from the time evolution equations of coupled moments. Nevertheless, sufficient flexibility has been built into the architecture of 2DX, that some kinetic effects can be incorporated into 2DX-compatible models. Two such examples are finite Larmor radius (FLR) effects and Landau damping.

Ion FLR effects are important when $k_{\perp}\rho_i \sim 1$ or larger. In the field-line-following coordinate system, the dominant contribution to k_{\perp} for large toroidal mode number n , is described by an eikonal phase factor. This factor gives rise to two input “profiles” k_b and k_{ψ} that (up to the multiplicative constant n) are a consequence of geometry alone.¹⁵ The remaining (non-eikonal-like) spatial variation of the eigenfunction is described by a slowly varying envelope function that is solved for numerically. Because FLR effects are of most interest for the high- n modes, it is sufficient in most cases to retain only the $k_{\perp}\rho_i$ resulting from k_b and k_{ψ} . The 2DX architecture allows the inclusion of full Bessel-function FLR effects depending on $k_{\perp}^2 \approx k_b^2 + k_{\psi}^2$ in the model equation set, since these have the status of input-profiles. While we have not exploited this capability in the present project, its availability for future applications is an important feature of the code.

Inclusion of parallel Landau damping effects presents a conceptually more difficult problem, since there is no equivalent rapidly varying phase factor in the parallel direction. Moreover, the parallel conductivity appearing in the evolution equation for parallel current (or parallel vector potential) involves the Z-function, which contains both the eigenfrequency and, formally, the parallel derivative operator. This type of structure was, initially, outside the scope of generalized eigenvalue models treatable by 2DX.

Therefore, some effort was been devoted in the project to find an workable extension.¹⁶ This was achieved with an iterative method which allows progressive refinements to the description of the parallel conductivity, based on an initial “effective” parallel wavenumber. Both the method, and a benchmark test, are described in more detail in Appendix J. The approach was motivated in part by earlier work that was foundational to the development of gyro-fluid models.¹⁷

IV. Benchmark-verification tests and debugging applications

A. Development of benchmark tests

In addition to code development, a significant part of the 2DX project was devoted to benchmarking-verification activities. These tests accomplish multiple purposes: (i) they verify the coding of the 2DX kernel, the physics models (structure files) and the handling geometry and input profiles (gridfiles) employed by 2DX, (ii) they simultaneously verify the other codes to which 2DX was compared, and (iii) they provide solid results which other codes can employ as benchmark targets. A summary of benchmark tests is provided in Table 1. Tests are described in the appendices as indicated by the lettered references; additional references from the bibliography are also given.

Of particular note are two tests: the resistive ballooning (RB) test in DIII-D divertor geometry, and the (ideal) ELM benchmark for (closed surface) tokamak geometry. A comparison of eigenmodes for the RB divertor geometry test is shown in Fig. 1. This test is important for establishing a correct treatment of the geometric quantities (e.g. metric coefficients, curvature, etc.) in realistic geometry, and for correct treatment of boundary and matching conditions in the various topological regions (e.g. toroidal periodicity in field-line-following coordinates, and matching across closed surfaces, main and private SOL).

In contrast the (ideal) ELM benchmark test employed a simpler geometry, but is significant because it is part of an ongoing effort to benchmark a number of codes on the same test case. The peeling-ballooning instability model being tested here is thought to describe the onset of Type I ELMs, which are of great interest for tokamak operation. The results, shown in Fig. 2, are for a cross-comparison of four codes (2DX, ELITE, GATO and BOUT++).¹⁸ Similar work^{19,20} has also benchmarked ELITE, GATO, BOUT++, and NIMROD in various combinations. Convergence studies of the 2DX code results are given in Appendix I.

Model	Geometry	Analytic tests	BOUT or BOUT++ comparison	Reference
RB	flux tube	X	X	5,9,10, B
RB	divertor		X	5
ES RD	slab	X	X	5,9,10, C
ES RD	cylinder		X	5
EM RD	slab	X	X	5,9,10, C
slab ITG	slab	X	X	5,9,10, D
toroidal ITG	flux tube	X		10, H
GAM	flux tube	X		5,9,10, E
EM kink	cylinder	X		10, F
ELM (ideal)	tokamak		X	10, I
parallel KH	slab	X		10, G
kinetic RB	flux tube	X		10, J

Table 1 2DX Benchmark tests

Notes and key:

Physics models tested:

RB = resistive ballooning

RD = resistive drift

ES = electrostatic

EM = electromagnetic

ITG = ion temperature gradient (slab and toroidal are distinct branches)

GAM = geodesic acoustic mode

KH = Kelvin Helmholtz

Geometries employed:

slab = Cartesian slab with one or more components of k specified as input

flux tube = model flux tube geometry for closed surfaces

divertor = global full X-point divertor geometry using experimental equilibria

cylinder = LAPD geometry

tokamak = model toroidal geometry for closed surfaces

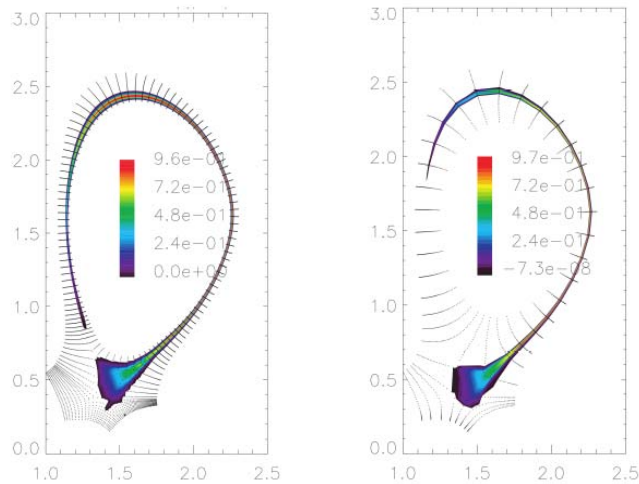


Fig. 1 Eigenmode structure from 2DX for the resistive ballooning mode in D-IIID edge geometry (left) and the same from BOUT (right). Growth rates for a variety of cases agree to well within the 5% measurement tolerance for exponential fits in BOUT. (see Ref. 5).

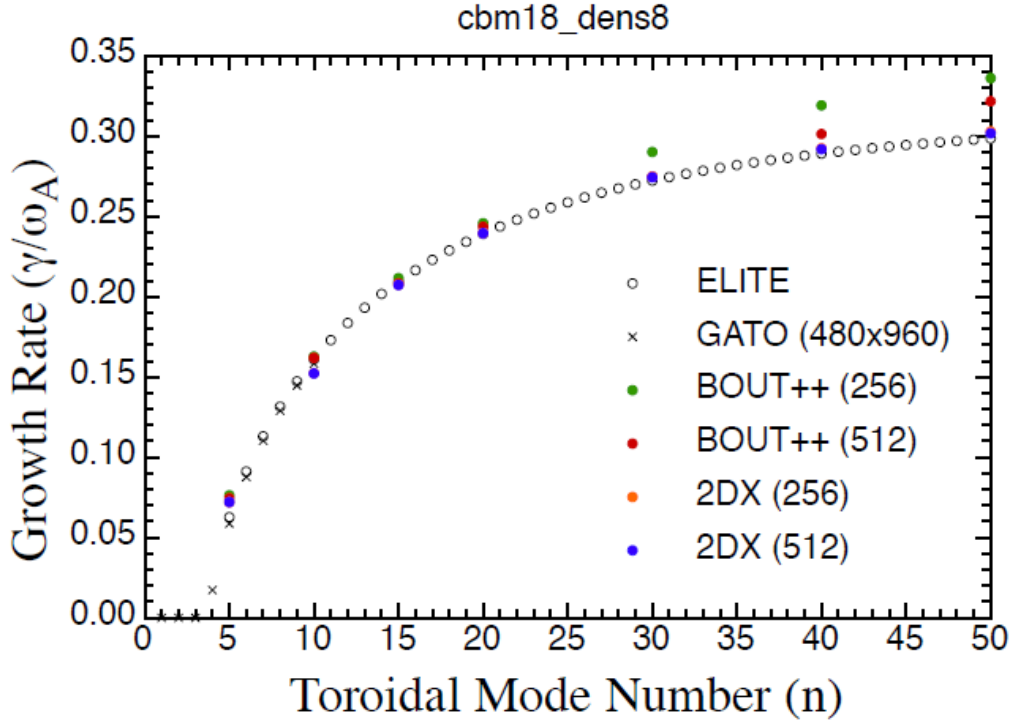


Fig. 2 Benchmark test results for the ideal MHD ELM case: Normalized growth rate as a function of toroidal mode number n at various resolutions. (Here the dimensionless frequency is defined using a $R/B = 1.5$ m/T Alfvén normalization.) The physics model is ideal MHD for ELITE and GATO and three-field reduced MHD containing peeling and ballooning terms for 2DX and BOUT++. A shifted annulus toroidal geometry is employed. All code results agree for low mode numbers but show slight disagreement for high mode numbers, where BOUT++ results are not converged. 2DX results do not show significant changes with resolution (256 case points are hidden), suggesting they are converged. [ELITE and GATO results, and final figure courtesy of P.B. Snyder]

B. Application of 2DX as a debugging aid

2DX and BOUT/BOUT++ are among very few codes that can describe boundary instabilities in full X-point divertor geometry. This fact alone makes 2DX an important tool for cross-code verification and debugging. A description of one such application follows.

The Resistive-Ballooning (RB) test case used for 2DX was applied to the BOUT++ code,⁸ which is the latest, enhanced and advanced, version of the BOUT code. However the initial results from BOUT++ showed disagreement with 2DX and the older version of the code, BOUT-06, for which this benchmark was successful.

On Apr. 24, 2011 a bug was finally found in BOUT++. After fixing this bug the RB growth rate values lie on the right curve (see Fig. 3). The bug was in setting a wrong coefficient in the off-diagonal terms of the Laplace operator, which physically corresponds to the integrated shear. That part of code was originally copied over from BOUT-06 but then some entropy was added to it by converting it to C++. Since this is related to fine details of real magnetic geometry of a tokamak, 2DX is one of very few numerical tools available that could help in finding this bug.

The previous linear ELM benchmarks with BOUT++¹⁹ did not reveal this problem because for ELM simulations a different option in the code was used for treating the geometry.

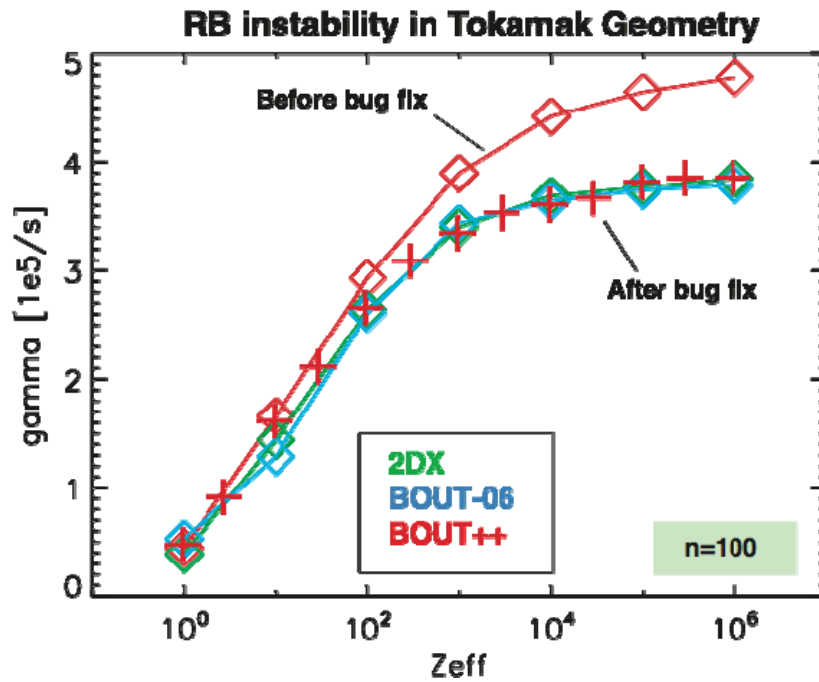


Fig. 3 Illustration of an application of 2DX as a debugging aid. The plot show the 2DX and BOUT-06 results in agreement for the resistive ballooning (RB) growth rates. BOUT++ was brought into agreement with the other codes after a bug fix related to real tokamak geometry (see text). 2DX is one of very few numerical tools available that could help in finding this bug.

V. Physics applications

The 2DX project has been successful in demonstrating new physics capabilities and applications in several areas of prime importance to the study of edge plasmas relevant to fusion. In particular, the project has:

- demonstrated a transition in the midplane-divertor parallel connection of resistive ballooning modes in NSTX. Disconnection occurs on flux surfaces near the separatrix in accordance with theoretical estimates^{11,12} and recent experimental GPI blob data.^{12,21} (see Fig. 4)
- carried out a detailed linear stability analysis²² of an Alcator C-Mod EDA edge plasma, which would have been impossible or extremely impractical without 2DX. A large number of physics models were explored, in a quest for a linear edge mode with the characteristics of the quasi-coherent mode. The absence of a strong candidate strengthens speculation on a possible role for the inverse cascade. (see Fig. 5)
- showed that the perpendicular KH instability in an EDA edge plasma is mitigated by both ion pressure and Alfvén coupling. The results are consistent with previous studies²³ and give a new perspective on the role of toroidal geometry.
- reproduced a linear benchmark for ELM growth rates in the ideal MHD model²⁴ (see Sec. IV and Appendix I). This work sets the stage for employing 2DX's capability for advanced physics models (extended MHD and multi-fluid physics) and full divertor geometry (including the SOL) to the study of ELMs.

Both the QC mode and ELM studies enabled by, and begun under, this Phase II grant are expected to continue under separate funding. They represent two prime applications of 2DX that exploit its full-geometry flexible-physics-model approach. Because of the importance of the QC and ELM modes to fusion plasmas, and the many physics questions that remain to be explored, 2DX is expected play an important and unique role in future edge research.

Finally, the application to disconnection in NSTX mentioned above illustrated another emerging application for 2DX: support for nonlinear studies that use the SOLT²⁵ 2D turbulence code. While SOLT is nonlinear, it makes analytical approximations in the parallel dynamics and in the geometry. On the other hand, 2DX, while linear, employs full geometry and parallel dynamics. These codes can be used synergistically to better understand the nature of the edge plasma. Furthermore they compliment codes such as BOUT and BOUT++ which have both nonlinear and full geometry capabilities, but can be numerically challenging to run in some cases of

interest. Comparative studies using all available codes, each with their own strengths and weaknesses, is almost certainly the best route to progress.

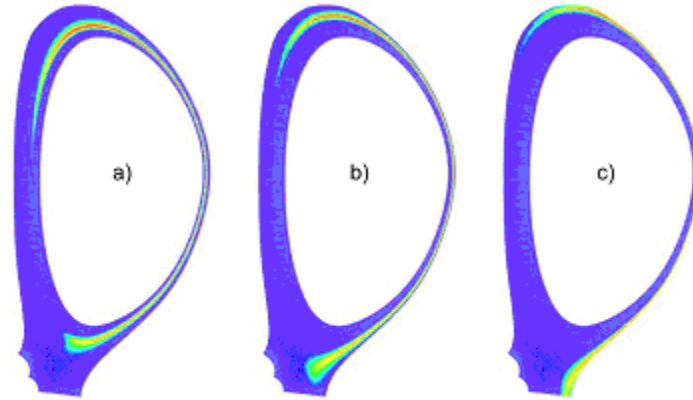


Fig. 4 Results of the 2DX NSTX connection length study for modes occurring a) inside the separatrix, b) at the separatrix, and c) in the SOL. Note that near the separatrix, modes are disconnected from the divertor plate.

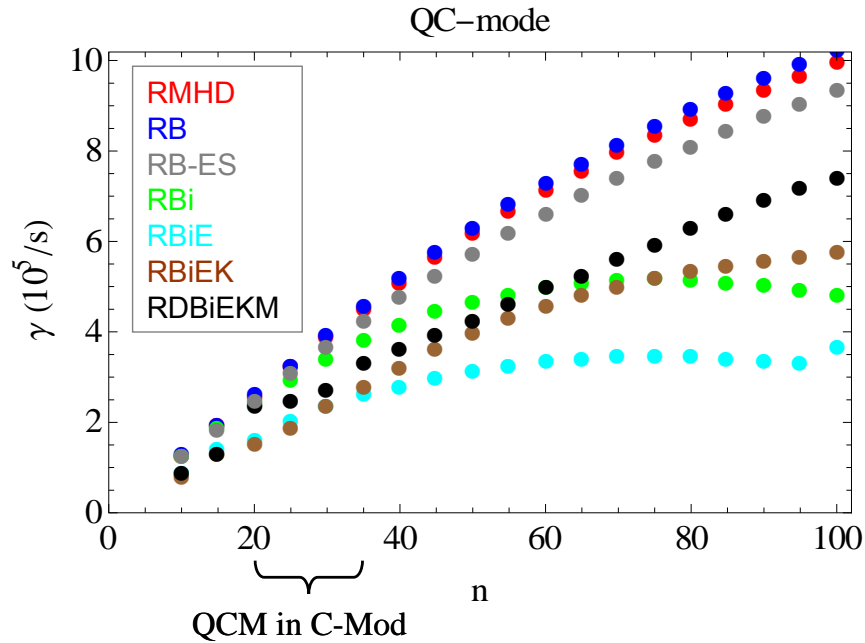


Fig. 5 Results of the 2DX linear stability analysis of an Alcator C-Mod EDA edge plasma. Shown are growth rates γ vs. toroidal mode number n for candidate quasi-coherent modes. Seven different physic models were employed ranging from RMHD = reduced resistive MHD to a “full” model (RDBiEKM) containing curvature-driven resistive and ideal ballooning, gradient-driven collisional and inertial drift waves, perpendicular flow-driven Kelvin-Helmholtz, ion diamagnetic current (FLR) and E_r shear. There is no sharp peak in growth rates for any of the models at the observed wavelength range in Alcator C-Mod experiments.

VI. Conclusions

As should be evident from the preceding summary, and the detailed appendices, the 2DX project has been very successful in developing a versatile and flexible computational tool for application to edge physics problems in magnetic fusion plasmas. The use of eigenvalue methods enables accurate, high resolution studies with modest computation effort, sometime several orders-of-magnitude less than time-stepping codes.⁵ The architecture has been designed to further the goal of error-free coding of a wide set of physics models. This has been achieved through encapsulation of the numerical methods, the physics models and the input-output (IO) in separate but integrated portions of coding, files, and auxiliary tools.⁵ Advanced fluid models have been coded and benchmarked for a large number of cases. A set of standardized benchmark tests has been developed and made available for future use.^{9,10} Finally, some interesting and unique physics applications have already been carried out, and many future applications of 2DX are envisaged.

In addition to linear code applications for growth rate benchmarking-verification and physics studies, several more far-reaching types of applications seem possible. One example is to post-process the 2DX output eigenmodes to construct second-order (“quasilinear”) particle, heat and momentum fluxes, and to study coupling coefficients for wave-wave interactions, inverse cascade and zonal flow generation in inhomogeneous full-geometry edge plasmas. Finally, the success of the 2DX approach demonstrated here has stimulated an extension of the method to higher dimensionality (in particular including velocity space) and more general topologies, that could implement a fully kinetic treatment.²⁶

Acknowledgments

This work was supported by the U.S. Department of Energy (DOE) under grant DE-FG02-07ER84718; however, such support does not constitute an endorsement by the DOE of the views expressed herein.

References

- ¹ K H Burrell, M E Austin, D P Brennan et al., Plasma Phys. Control. Fusion **44**, A253 (2002).
- ² M. Greenwald, et al., Phys. Plasmas **6**, 1943 (1999).
- ³ S. J. Zweben, J. A. Boedo, O. Grulke, C. Hidalgo, B. LaBombard, R. J. Maqueda, P. Scarin, and J. L. Terry, Plasma Phys. Controlled Fusion **49**, S1 (2007).
- ⁴ B. A. Carreras, J. Nucl. Mater. **337-339**, 315 (2005).
- ⁵ D. A. Baver, J. R. Myra and M.V. Umansky, Comp. Phys. Comm. **182**, 1610, (2011); also Appendix **K** of this report.
- ⁶ M.V. Umansky, X.Q. Xu, B. Dudson, L.L. LoDestro, J.R. Myra, Comp. Phys. Comm. **180**, 887 (2009).
- ⁷ X.Q. Xu, et al., Phys. Plasmas **7** (2000) 1951.
- ⁸ B.D. Dudson, M.V. Umansky, X.Q. Xu, P.B. Snyder and H.R. Wilson, Comp. Phys. Comm. **180**, 1467 (2009).
- ⁹ The ECC site is presently at <http://www.lehigh.edu/~inecc/> however the host for this site may change. 2DX benchmark tests have been submitted for posting.
- ¹⁰ <http://www.lodestar.com/research/vnv/>
- ¹¹ S. I. Krasheninnikov, D. A. D'Ippolito, and J. R. Myra, (Theory Review) J. Plasma Physics **74**, 679 (2008).
- ¹² D.A. D'Ippolito, J.R. Myra and S.J. Zweben, (Theory and Experimental Review) Phys. Plasmas **18**, 060501 (2011).
- ¹³ see Refs. **5**, **10** and Appendices **B – J** of this report.
- ¹⁴ A. Zeiler, J.F. Drake, and B. Rogers, Phys. Plasmas **4**, 2134 (1997).
- ¹⁵ for a notation-compatible exposition of the eikonal method in toroidal geometry see e.g. J. R. Myra, D.A. D'Ippolito, X.Q. Xu and R.H. Cohen, Phys. Plasmas **7**, 4622 (2000).
- ¹⁶ D. A. Baver, J. R. Myra and M.V. Umansky, "Gyrofluid model implementation using the 2DX eigenvalue code," Bull. APS **55**, 49 (2010), paper BP9-103.
- ¹⁷ G.W. Hammett and F.W. Perkins, Phys. Rev. Lett. **64**, 3019 (1990).
- ¹⁸ ELITE and GATO results, courtesy of P.B. Snyder
- ¹⁹ B. D. Dudson, X. Q. Xu, M. V. Umansky, H. R. Wilson and P. B. Snyder, Plasma Phys. Control. Fusion **53**, 054005 (2011).
- ²⁰ B. J. Burke, S. E. Kruger, C. C. Hegna, P. Zhu, P. B. Snyder, C. R. Sovinec, and E. C. Howell, Phys. Plasmas **17**, 032103 (2010).
- ²¹ R.J. Maqueda, R. Maingi, NSTX team, Phys. Plasmas **16**, 056117 (2009).
- ²² J.R. Myra, D. A. Baver and M.V. Umansky, "Linear Stability Analysis of an EDA H-mode Edge Plasma: A Quest for the Quasi-Coherent Mode," presented at the *Transport Task Force Workshop*, San Diego, CA, April 6-9, 2011.
- ²³ B. N. Rogers and W. Dorland, Phys. Plasmas **12**, 062511 (2005).
- ²⁴ D. A. Baver, J.R. Myra and M.V. Umansky, "ELM modeling using the 2DX eigenvalue code," presented at the *Transport Task Force Workshop*, San Diego, CA, April 6-9, 2011.
- ²⁵ D. A. Russell, J. R. Myra, and D. A. D'Ippolito, Phys. Plasmas **16**, 122304 (2009).
- ²⁶ ArbiTER Phase I SBIR grant, D.A. Baver (PI).

Six-field-model equations

J. R. Myra, D. A. Baver

Lodestar Research Corp., 2400 Central Ave. P-5, Boulder, Colorado 80301

M. V. Umansky

Lawrence Livermore National Laboratory, Livermore, CA 94550

A. Introduction

The equations given below comprise the linearized six-field fluid model, which provides time advancement equations for vorticity (electrostatic potential), density, parallel velocity, electron temperature, ion temperature and parallel vector potential (parallel current). The equation set is essentially a linearized version of the BOUT six-field model,^{1,2} and is similar to that employed by other authors.³ The present model allows for equilibrium parallel current gradients (the kink-peeling mode drive), and equilibrium flow gradients (perpendicular and parallel Kelvin-Helmholtz drive, as well as shear-flow suppression effects on other instabilities).

The equations are in dimensionless Bohm units, i.e. time scales are normalized to a reference inverse ion cyclotron frequency Ω_{ir} and length scales to $\rho_{sr} = c_{esr}/\Omega_{ir}$ where $c_{esr}^2 = T_{er}/m_i$. The normalized density, and temperatures are n/n_{er} , T_e/T_r and T_i/T_r where we choose a single reference temperature for both electrons and ions.

In a few cases, we drop terms small in $1/k_{\perp}L_{\perp}$ where k_{\perp} is the perpendicular (to B) wavenumber and L_{\perp} an equilibrium scale length. This is the case for the gyroviscous terms in the vorticity equation, which are usually only important at high k_{\perp} . Similarly, the magnetic field in the curvature is not perturbed because it would be small in $1/(k_{\perp}R)$ compared with the perturbing the fields that the curvature operates on.

This physics model, when solved as an eigenvalue problem for mode growth rates, includes familiar edge modes such as the drift wave, the resistive ballooning mode, the ion temperature gradient mode, edge peeling and kink modes, sheath-driven modes, perpendicular and parallel Kelvin-Helmholtz modes, and many others. Collisionality, ion diamagnetism, background flows, electromagnetism and many other effects are included in this physics description. Every important edge and scrape-off-layer mode discussed in the contemporary literature is embedded in this model.

Coding of this equation set into a “structure” file, and subsequent checking was greatly facilitated by use of the equation language, and equation language viewer. In practice, since this equation set contains essentially all terms that are envisaged to be of

interest, most have which have been benchmarked, subset models are most easily obtained by starting from the present model, and commenting out unneeded terms.

B. Six-field model equations

The equations for the perturbed field of electrostatic potential $\delta\Phi$, density δn , parallel velocity δu , electron temperature δT_e , ion temperature δT_i , and parallel vector potential δA are

$$\begin{aligned} \gamma\delta\varpi = & -\mathbf{v}_E \cdot \nabla\delta\varpi - u\nabla_{\parallel}\delta\varpi - \delta\mathbf{v}_E \cdot \nabla\varpi + \frac{2\mathbf{B}}{n}C_r\delta p - \frac{1}{n}C_g\delta n \\ & + \frac{\mathbf{B}^2}{n}\partial_{\parallel}\delta J_{\parallel} + \frac{\mathbf{B}^2}{n}\delta\mathbf{b} \cdot \nabla J_{\parallel} + \Gamma\nabla_{\perp}^2\delta\Phi + \mu_{ii}\nabla_{\perp}^4\delta\Phi \end{aligned} \quad (1)$$

$$\gamma\delta n = -\mathbf{v}_E \cdot \nabla\delta n - u\nabla_{\parallel}\delta n - \delta\mathbf{v}_E \cdot \nabla n + \frac{2}{\mathbf{B}}(C_r\delta p_e - nC_r\delta\Phi) - n\partial_{\parallel}\delta u + \partial_{\parallel}\delta J \quad (2)$$

$$\begin{aligned} \gamma\delta u = & -\mathbf{v}_E \cdot \nabla\delta u - u\nabla_{\parallel}\delta u - \delta\mathbf{v}_E \cdot \nabla u - \frac{1}{n}\nabla_{\parallel}\delta p - \frac{1}{n}\delta\mathbf{b} \cdot \nabla p - \frac{2T_i}{\mathbf{B}}C_r\delta u \\ & + \nabla_{\parallel}\mu_{ii}\partial_{\parallel}\delta u \end{aligned} \quad (3)$$

$$\begin{aligned} \gamma\delta T_e = & -\mathbf{v}_E \cdot \nabla\delta T_e - u\nabla_{\parallel}\delta T_e - \delta\mathbf{v}_E \cdot \nabla T_e + \frac{2(1.71)T_e}{3n}\partial_{\parallel}\delta J - \frac{2}{3}T_e\partial_{\parallel}\delta u \\ & + \frac{2}{3}\partial_{\parallel}\chi_{\parallel}(\nabla_{\parallel}\delta T_e + \delta\mathbf{b} \cdot \nabla T_e) + \frac{4T_e}{3\mathbf{B}}\left(\frac{1}{n}C_r\delta p_e - C_r\delta\Phi + \frac{5}{2}C_r\delta T_e\right) \end{aligned} \quad (4)$$

$$\begin{aligned} \gamma\delta T_i = & -\mathbf{v}_E \cdot \nabla\delta T_i - u\nabla_{\parallel}\delta T_i - \delta\mathbf{v}_E \cdot \nabla T_i + \frac{2T_i}{3n}\partial_{\parallel}\delta J - \frac{2}{3}T_i\partial_{\parallel}\delta u \\ & + \frac{4T_i}{3\mathbf{B}}\left(\frac{1}{n}C_r\delta p_e - C_r\delta\Phi - \frac{5}{2}C_r\delta T_i\right) \end{aligned} \quad (5)$$

$$\begin{aligned} \gamma\left(\frac{n}{\delta_{er}^2} - \nabla_{\perp}^2\right)\delta A = & -\mathbf{v}_E \cdot \nabla\left(\frac{n}{\delta_{er}^2} - \nabla_{\perp}^2\right)\delta A - u\nabla_{\parallel}\left(\frac{n}{\delta_{er}^2} - \nabla_{\perp}^2\right) + v_e\nabla_{\perp}^2\delta A - \mu n\nabla_{\parallel}\delta\Phi \\ & + \mu T_e\nabla_{\parallel}\delta n + T_e\mu\delta\mathbf{b} \cdot \nabla n + 1.71\mu n\nabla_{\parallel}\delta T_e + 1.71n\mu\delta\mathbf{b} \cdot \nabla T_e \end{aligned} \quad (6)$$

where

$$\delta\varpi = \nabla_{\perp}^2\delta\Phi + \frac{1}{n}\nabla_{\perp}^2\delta p_i \quad (7)$$

$$\varpi = \nabla_{\perp}^2\Phi + \frac{1}{n}\nabla_{\perp}^2 p_i \quad (8)$$

In the above, for any perturbed field variable δX

$$\mathbf{v}_E \cdot \nabla \delta X = \frac{i}{B} k_b (\partial_r \Phi) \delta X = i k_b v_E \delta X \quad (9)$$

where k_b is the binormal component of the wavevector, related to the toroidal mode number n_ζ by

$$k_b = -n_\zeta \frac{B}{RB_\theta} \quad (10)$$

and v_E is the equilibrium $\mathbf{E} \times \mathbf{B}$ velocity. Additionally

$$v_e = 0.51 \frac{n}{T^{3/2}} v_r \quad (11)$$

where v_r is the electron-ion collision frequency at the reference density and temperature

$$\mu = \frac{m_i}{m_e} \quad (12)$$

$$\delta p = T_e \delta n + n \delta T_e + T_i \delta n + n \delta T_i \equiv \delta p_e + \delta p_i \quad (13)$$

$$\partial_{\parallel} \delta X = B \nabla_{\parallel} (\delta X / B) \quad (14)$$

In slab geometry, for any equilibrium profile Q , $\partial_r Q$ is the usual radial derivative and in flux coordinates

$$\partial_r Q = RB_p \frac{dQ}{d\psi} \quad (15)$$

The curvature operator is given by

$$C_r = \mathbf{b} \times \boldsymbol{\kappa} \cdot \nabla \quad (16)$$

and involves only the radial derivative operator (since the binormal component of the derivative is determined by the toroidal mode number). C_g allows for other forces, e.g. centrifugal forces due to $\mathbf{E} \times \mathbf{B}$ rotation at speed v_E results in

$$C_g = i k_b \partial_r (v_E^2 / 2) \quad (17)$$

which is analogous to the curvature term.

The perturbed vector potential δA is not strictly Bohm-normalized. Rather we define

$$\delta A \equiv \frac{\delta A_{\parallel}}{\beta_r} = \mu \delta_{er}^2 \delta A_{\parallel} \quad (18)$$

where δA_{\parallel} is the Bohm-normalized quantity. Here β_r is the reference electron-plasma beta and δ_{er} is the reference skin-depth

$$\delta_{\text{er}} = \left(\frac{c}{\omega_{\text{pe}} \rho_s} \right)_r \equiv \left(\frac{\delta_e}{\rho_s} \right)_r \quad (19)$$

Then Ampere's law becomes

$$\delta J = -\nabla_{\perp}^2 \delta A \quad (20)$$

The perturbed magnetic field terms contain quantities arising from

$$\nabla_{\parallel} \delta \mathbf{X} \rightarrow \nabla_{\parallel} \delta \mathbf{X} + \delta \mathbf{b} \cdot \nabla \mathbf{X} \quad (21)$$

where

$$\delta \mathbf{b} \cdot \nabla \mathbf{Q} = i \frac{k_b (\partial_r \mathbf{Q})}{\mu \delta_{\text{er}}^2 B} \delta A \quad (22)$$

In addition to the terms shown explicitly above, we have coded and experimented with (but not used much) hyper-diffusion terms of the form

$$\gamma \delta \mathbf{X}_j = -H_j \delta \mathbf{X}_j + \dots \quad (23)$$

where $\delta \mathbf{X}_j$ is a field variable and

$$H_j = H_{\perp j} \nabla_{\perp}^4 + H_{\parallel j} \nabla_{\parallel}^4 \quad (24)$$

Such terms are potentially useful in suppressing short-wavelength grid-scale modes which are not represented accurately on a given mesh.

C. Boundary conditions

In the radial direction, boundary conditions (BCs) are controlled by a parameter Λ_x . Both zero-value and zero-derivative boundary conditions are supported by 2DX.

$$\Lambda_x = \begin{cases} 0, & \text{0 value BC} \\ 1, & \text{0 derivative BC} \end{cases} \quad (25)$$

Normally, (i.e. when the radial domain is large enough to contain the eigenfunctions of interest) results are insensitive to the choice of Λ_x . The option of zero-derivative BCs is useful for benchmarking against local theory (where the code is essentially run on a single flux tube).

In the parallel direction, BCs are phase-shift periodic on closed surfaces, i.e. they obey physical periodicity conditions in both poloidal and toroidal directions. In field-line following coordinates, this implies

$$\delta \mathbf{X}|_{y=0} = \delta \mathbf{X}|_{y=2\pi} \exp(-2\pi i n_{\zeta} q) \quad (26)$$

where n_{ζ} is the toroidal mode number and q is the safety factor.

On open field lines, we typically impose BCs that correspond to termination on conducting or insulating end-plates, or to sheath BCs. This is achieved by introducing several switching parameters Λ_1 , Λ_2 , and Λ_3 which are normally set to 0, 1, or “ ∞ ”. The explicit form of the parallel BCs is

$$-\nabla_{\perp}^2 \delta A = -\sigma n (T_e + T_i)^{1/2} \left(\frac{1}{2} \frac{\delta T_e + \delta T_i}{T_e + T_i} - \frac{1}{2} \frac{\delta T_e}{T_e} + \Lambda_1 \frac{\delta \Phi}{T_e} + \frac{(\Phi + \Lambda_2) \delta T_e}{T_e^2} \right) \quad (27)$$

$$\delta u = -\sigma \frac{\delta T_e + \delta T_i}{2(T_e + T_i)^{1/2}} \Lambda_3 \quad (28)$$

$$S_E \left[\delta T_e + \frac{\delta T_e + \delta T_i}{2(1 + T_i/T_e)} \right] + \delta \Phi = \frac{\sigma \chi_{\parallel} (\nabla_{\parallel} \delta T_e + \delta \mathbf{b} \cdot \nabla T_e)}{(T_i + T_e)^{1/2}} \quad (29)$$

where

$$\sigma = \frac{\mathbf{b} \cdot \mathbf{n}}{|\mathbf{b} \cdot \mathbf{n}|} = \pm 1 \quad (30)$$

and \mathbf{n} is the unit normal to the surface, pointing into the plasma. For example, neglecting perturbed temperatures in the parallel current BC Eq. (27), Λ_1 can take the values $\Lambda_1 = 1$ (sheath BC), $\Lambda_1 = \infty$ (metal wall BC), $\Lambda_1 = 0$ (insulating BC). Sheath BCs which self-consistently adjust to the correct limit in all three equations are given by the combination $\Lambda_1 = 1$, $\Lambda_2 = 0$, $\Lambda_3 = 1$. In the above, S_E (the sheath energy transmission coefficient) is taken as a constant. ($S_E = 2$ for a Maxwellian plasma so that the total electron energy removed from the plasma including the $\delta \Phi$ term is about $2T_e + 3T_e = 5T_e$.)

References

1. M.V. Umansky, X.Q. Xu, B. Dudson, L.L. LoDestro, J.R. Myra, *Comp. Phys. Comm.* **180**, 887 (2009).
2. X.Q. Xu, et al., *Phys. Plasmas* **7** (2000) 1951.
3. A. Zeiler, J.F. Drake, and B. Rogers, *Phys. Plasmas* **4**, 2134 (1997).

Resistive ballooning test of the 2DX code

D. A. Baver and J. R. Myra

Lodestar Research Corporation, Boulder Colorado 80301

Maxim Umansky

Lawrence Livermore National Laboratory

1 Introduction

This test was devised to verify the ability of the 2DX eigenvalue code to correctly solve a simple fluid model relevant to edge turbulence in tokamaks. Since the functionality of the 2DX code depends on both the source code itself and the input file defining the system of equations to solve (structure file), this test demonstrates both. Since a similar test was performed on an earlier version of 2DX, this verifies that the current version retains this functionality. Moreover, since the structure file for this test represents a subset of a more general 6-field model, many of the terms in that test are also verified.

This test compares 2DX results to BOUT simulations, and to approximate analytic solutions in the limits of large and small binormal wavenumber.

2 Description

2.1 Code structure

The 2DX code is a highly flexible eigenvalue solver designed for problems relevant to edge physics in toroidal plasma devices. Its flexibility stems from the use of a specialized input file containing instructions on how to set up a particular set of equations. Because of this, the 2DX code permits model equations to be changed without altering its source code. The drawback to this approach is that any change to the structure file represents a potential source of error, necessitating re-verification. This problem is offset by the fact that the source code remains unchanged, thus testing one structure file builds confidence in the underlying code that interprets the structure file. Also, structure files can be translated into analytic form, thus allowing the user to verify that the file contains the equations intended.

The structure file contains two main parts: an elements section, which constructs the differential operators and other functions used in a particular set

of equations, and a formula section, which assembles these into an actual set of equations. This separation means that elements can be recycled in other structure files. By testing one structure file, one builds confidence in the elements used in that file. The main source of error when switching to a different structure file then is in the formula section, which can be manually verified by translating into analytic form.

Regardless of the content of the structure file, the 2DX code is fundamentally a finite-difference eigenvalue solver. As such, it is subject to the limitations of any code of its type.

2.2 Model equations

For this test we use the following model equations [1]-[3]:

$$\gamma \nabla_{\perp}^2 \delta \Phi = + \frac{2B}{n} C_r \delta p - \frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (1)$$

$$\gamma \delta n = -\delta v_E \cdot \nabla n \quad (2)$$

$$-\gamma \nabla_{\perp}^2 \delta A = \nu_e \nabla_{\perp}^2 \delta A - \mu n \nabla_{\parallel} \delta \Phi \quad (3)$$

where:

$$\delta p = (T_e + T_i) \delta n + n(\delta T_e + \delta T_i) \quad (4)$$

$$C_r = \mathbf{b} \times \kappa \cdot \nabla = -\kappa_g R B_p \partial_x + i(\kappa_n k_z - \kappa_g k_{\psi}) \quad (5)$$

$$\nabla_{\perp}^2 = -k_z^2 - jB(k_{\psi} - i\partial_x R B_p)(1/jB)(k_{\psi} - iR B_p \partial_x) \quad (6)$$

$$\partial_{\parallel} Q = B \nabla_{\parallel} (Q/B) \quad (7)$$

$$\nabla_{\parallel} = j \partial_y \quad (8)$$

$$\delta v_E \cdot \nabla Q = -i \frac{k_z (R B_p \partial_x Q)}{B} \delta \Phi \quad (9)$$

$$\nu_e = .51 \nu_r n / T_e^{3/2} \quad (10)$$

2.3 Boundary conditions

This test case uses phase-shift periodic boundary conditions in the parallel direction, and zero-derivative boundary conditions in the radial direction. The phase shift in the parallel direction is given by:

$$e^{i2\pi n q} \quad (11)$$

2.4 Profile setup

The formulas in Eq. 1-3 are normalized to Bohm units. Distances are measured in units of ρ_s and time is measured in units of ω_{ci}^{-1} , with ρ_s and ω_{ci} calculated at reference values of n_e , T_e , and B . Profiles of n_e , T_e , and B are provided as multiples of these reference values. Output eigenvalues are multiplied by ω_{ci} . Resistivity is given by the formula:

$$\nu_r = \frac{\mu}{.51\sigma} \quad (12)$$

where

$$\sigma = 1.96 \frac{\omega_{ce}}{\nu_{ei}} \quad (13)$$

The geometry used is an idealized toroidal annulus with major radius R , minor radius a , and thickness δa . The density profile is exponential with scale length L_n , and temperature profiles are flat. Curvature is assumed, and is given by:

$$\kappa_n = \frac{\cos(y)}{R} \quad (14)$$

The function q may be sheared, but shear is set to zero for the test case given. The value of this constant q is given in Sec. 4.

Parallel derivatives are calculated using the Jacobian factor $j = 1/qR$. Toroidal mode number is calculated by $n = k_z a/q_0$.

3 Analytic results

The solutions to the equation set in Eq. 1-3 can be solved by first assuming that $\mu \gg 1$, in which case the equations can be reduced to the form:

$$\partial_y^2 \delta\Phi + \frac{\alpha}{\gamma} (\gamma_0^2 \cos(y) - \gamma^2) \delta\Phi = 0 \quad (15)$$

where:

$$\alpha = \frac{q^2 R^2 k_z^2}{\sigma} \quad (16)$$

$$\gamma_0 = \sqrt{\frac{2}{RL_n}} \quad (17)$$

For the limit where k_z is very large, we can use the approximation $\cos(y) \approx 1 - y^2/2$ to get a Hermite equation. This results in the approximate solution:

$$\gamma = \omega_{ci} \left(\gamma_0 - \frac{\sqrt{\frac{\alpha\gamma_0}{8}}}{\alpha} \right) \quad (18)$$

For the limit where k_z is very small, a more complicated approximation can be used. This arises from the ansatz $\delta\Phi = e^{inq\theta} + Ae^{i\theta(nq+1)}$. This yields the analytic formula:

$$\frac{\alpha^2\gamma_0^4}{2\gamma(1+2nq)} = \alpha\gamma + n^2q^2 \quad (19)$$

4 Numerical results

The code was tested by sweeping the variable k_z from .001 to 1000 cm and plotting the fastest growing eigenvalue. The parameters used in this test are shown in table 1:

$$\begin{aligned} a &= .75 \text{ cm} \\ \delta a &= .3 \text{ cm} \\ R &= 207.5 \text{ cm} \\ L_n &= 9.4 \text{ cm} \\ Z_{eff} &= 32 \\ B &= 1 \text{ T} \\ n_e &= 10^{14} \text{ cm}^{-3} \\ m_i/m_p &= 2 \\ \mu &= 10^4 \\ T_i &= 1 \text{ eV} \\ T_e &= 100 \text{ eV} \\ \ln\Lambda &= 12.4 \end{aligned}$$

The results of this test are shown in Fig. 1. The red lines near the ends of the plot are analytic solutions. The blue dashed line is the asymptotic limit of the analytic solution for high k_z . The green dots are solutions from 2DX, whereas the black dots are solutions from a previous version of 2DX using a 2-field model. The blue crosses are simulation results from BOUT. In addition, a table of the raw eigenvalue data is shown in table 2.

References

- [1] H. Strauss, Phys. Fluids **24**, 2004 (1981).
- [2] T. C. Hender, B. A. Carreras, W. A. Cooper, J. A. Holmes, P. H. Diamond, and P. L. Similon, Phys. Fluids **27**, 1439 (1984).

- [3] P. N. Guzdar and J. F. Drake, *Phys. Fluids B* **5**, 3712 (1993).

nx	4	ny	16
dx	.693242	dy	.392699
γ	$\gamma(s^{-1})/4.79 \times 10^7$	n	.5 $k_z(cm^{-1})$
μ	10^4	ν_r	.00131267
q	1.5	k_z	-.144249 $k_z(cm^{-1})$
j	.000474896	k_ψ	0
κ_n	.00711444 $\cos(y)$	κ_g	0
B	1	RB_p	1
n_0	$e^{-x/65.1647}$	T_e	1

Table 1: Non-dimensional profile functions and parameters used in the resistive ballooning test case, as a function of the dimensional input $k_z(cm^{-1})$.

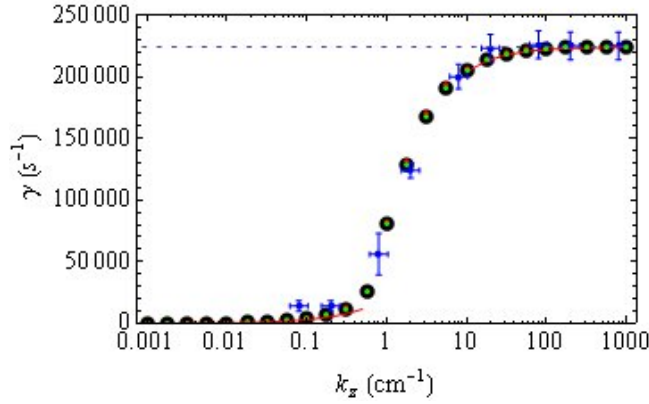


Figure 1: Growth rate vs. k_z for the resistive ballooning model. Green dots are solutions from 2DX, black dots are solutions from 2DX from a 2-field model, blue crosses are BOUT results, and red lines are analytic results.

$k_z(cm^{-1})$	$\gamma(s^{-1})$	$k_z(cm^{-1})$	$\gamma(s^{-1})$
.001	7.65881	1.77283	127813
.00177283	13.7556	3.16228	167064
.00316228	792.258	5.62341	191107
.00562341	3890	10	205355
.01	3812.59	17.7283	213607
.0177283	3880.01	31.6228	218291
.0316228	4026.53	56.2341	220651
.0562341	4431.82	100	221637
.1	5429.65	177.283	222045
.177283	7606.79	316.228	222220
.316228	12267.2	562.341	222284
.562341	24588.4	1000	222305
1	80744.4		

Table 2: Growth rate vs. k_z for the resistive ballooning model.

Resistive drift wave test of the 2DX code

D. A. Baver and J. R. Myra

Lodestar Research Corporation, Boulder Colorado 80301

Maxim Umansky

Lawrence Livermore National Laboratory

1 Introduction

This test was devised to verify the ability of the 2DX eigenvalue code to correctly solve a simple fluid model relevant to edge turbulence in tokamaks. Since the functionality of the 2DX code depends on both the source code itself and the input file defining the system of equations to solve (structure file), this test demonstrates both. Since a similar test was performed on an earlier version of 2DX, this verifies that the current version retains this functionality. Moreover, since the structure file for this test represents a subset of a more general 6-field model, many of the terms in that test are also verified.

This test compares 2DX results to an exact analytic solution for the model equations of interest.

2 Description

2.1 Code structure

The 2DX code is a highly flexible eigenvalue solver designed for problems relevant to edge physics in toroidal plasma devices. Its flexibility stems from the use of a specialized input file containing instructions on how to set up a particular set of equations. Because of this, the 2DX code permits model equations to be changed without altering its source code. The drawback to this approach is that any change to the structure file represents a potential source of error, necessitating re-verification. This problem is offset by the fact that the source code remains unchanged, thus testing one structure file builds confidence in the underlying code that interprets the structure file. Also, structure files can be translated into analytic form, thus allowing the user to verify that the file contains the equations intended.

The structure file contains two main parts: an elements section, which constructs the differential operators and other functions used in a particular set

of equations, and a formula section, which assembles these into an actual set of equations. This separation means that elements can be recycled in other structure files. By testing one structure file, one builds confidence in the elements used in that file. The main source of error when switching to a different structure file then is in the formula section, which can be manually verified by translating into analytic form.

Regardless of the content of the structure file, the 2DX code is fundamentally a finite-difference eigenvalue solver. As such, it is subject to the limitations of any code of its type.

2.2 Model equations

For the resistive drift wave test, two different model equations are used [1]-[2]. One of these models is electrostatic, the other includes electromagnetic effects.

The electrostatic case uses the following equations:

$$\gamma \nabla_{\perp}^2 \delta \Phi = -\frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (1)$$

$$\gamma \delta n = -\delta v_E \cdot \nabla n \quad (2)$$

$$-\gamma \nabla_{\perp}^2 \delta A = \nu_e \nabla_{\perp}^2 \delta A - \mu n \nabla_{\parallel} \delta \Phi + \mu T_e \nabla_{\parallel} \delta n \quad (3)$$

where

$$\delta p = (T_e + T_i) \delta n + n(\delta T_e + \delta T_i) \quad (4)$$

$$C_r = \mathbf{b} \times \kappa \cdot \nabla = -\kappa_g R B_p \partial_x + i(\kappa_n k_b - \kappa_g k_{\psi}) \quad (5)$$

$$\nabla_{\perp}^2 = -k_b^2 - jB(k_{\psi} - i\partial_x R B_p)(1/jB)(k_{\psi} - iR B_p \partial_x) \quad (6)$$

$$\partial_{\parallel} Q = B \nabla_{\parallel} (Q/B) \quad (7)$$

$$\nabla_{\parallel} = j \partial_y \quad (8)$$

$$\delta v_E \cdot \nabla Q = -i \frac{k_z (R B_p \partial_x Q)}{B} \delta \Phi \quad (9)$$

$$\nu_e = .51 \nu_r n / T_e^{3/2} \quad (10)$$

The electromagnetic model uses the following alternate equation set:

$$\gamma \nabla_{\perp}^2 \delta \Phi = -\frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (11)$$

$$\gamma \delta n = -\delta v_E \cdot \nabla n \quad (12)$$

$$\gamma \left(\frac{n}{\delta_{er}^2} - \nabla_{\perp}^2 \delta A \right) = \nu_e \nabla_{\perp}^2 \delta A - \mu n \nabla_{\parallel} \delta \Phi + \mu T_e \nabla_{\parallel} \delta n + \mu T_e \delta \mathbf{b} \times \nabla n \quad (13)$$

where δ_{er} is the Bohm-normalized value of skin depth for the reference parameters and:

$$\delta \mathbf{b} \times \nabla Q = \frac{ik_b(\partial_r Q)}{\mu\delta_{er}^2 b} \delta A \quad (14)$$

2.3 Boundary conditions

This test case uses phase-shift periodic boundary conditions in the parallel direction, and zero-derivative boundary conditions in the radial direction. The phase shift in the parallel direction is given by:

$$e^{i2\pi nq} \quad (15)$$

For this particular case, q is set to an integer so as to establish a simple periodic domain.

2.4 Profile setup

The formulas in Eq. 1-13 are normalized to Bohm units. Values are converted by dividing input distances by ρ_s , and input magnetic fields are in Tesla. Output eigenvalues are multiplied by ω_{ci} . Resistivity is given by the formula:

$$\nu_r = \frac{\mu_u}{.51\sigma} \quad (16)$$

where

$$\sigma = 1.96 \frac{\omega_{ce}}{\nu_{ei}} \quad (17)$$

The geometry used is a periodic slab. Curvature effects are included in the equation set, but curvature is set to zero. Zero-derivative boundary conditions are used in the radial direction, and the domain is set to only two grid cells wide in that direction. This is done so as to approximate a 1-D problem using a 2-D code, and because the 2DX code cannot simulate domains that are only one grid cell wide in either direction.

The Jacobian factor used to calculate parallel derivatives is set so as to fix the parallel wavenumber of the fundamental mode of the system. As we will see later, the fundamental mode may or may not be the fastest growing mode depending on what wavenumber is selected. Setting the wavenumber of the fundamental mode is accomplished by simply having parallel positions range from π to $-\pi$ and setting the Jacobian equal to k_{\parallel} .

3 Analytic results

Since both the electrostatic and electromagnetic resistive drift wave models are tested in homogenous domains, they can be solved analytically by taking a Fourier transform in both directions. This allows the systems of differential equations to be reduced to algebraic matrix equations. Assuming that $\nabla_{\perp}^2 = -k_b^2$, this yields an eigenvalue problem of the form $Ax = \gamma x$, where A for the electrostatic model is:

$$\begin{bmatrix} 0 & -ik_b RB_p n / BL_n & 0 \\ 0 & 0 & -iBk_{\parallel} / n \\ ik_{\parallel} T / \beta_r \delta_{er}^2 k_b^2 & -ik_{\parallel} n / \beta_r \delta_{er}^2 k_b^2 & -.51\nu_r n / T^{3/2} \end{bmatrix} \quad (18)$$

and for the electromagnetic model:

$$\begin{bmatrix} 0 & -ik_b RB_p n / BL_n & 0 \\ 0 & 0 & -iBk_{\parallel} / n \\ \frac{ik_{\parallel} T / \beta_r n}{1 + \delta_{er}^2 k_b^2 / n} & \frac{-ik_{\parallel} 1 / \beta_r}{1 + \delta_{er}^2 k_b^2 / n} & \frac{-iTk_b RB_p / BL_n - .51\delta_{er}^2 \nu_r k_b^2 / T^{3/2}}{1 + \delta_{er}^2 k_b^2 / n} \end{bmatrix} \quad (19)$$

These can be solved using standard eigenvalue solving routines. The results of this calculation are shown in Figs. 1-2 along with the numerical results from 2DX.

4 Numerical results

The code was tested by sweeping the dimensionless variable σ_{\parallel} from .1 to 100 and plotting the fastest growing eigenvalue. In addition, the fastest growing value at the fundamental wavenumber was also calculated. The parameters used in this test are as follows:

$$\begin{aligned} \delta_{er}^2 &= 4 \\ \beta_r &= .02 \\ \nu_r &= .05 \\ RB_p &= 1 \\ k_b &= 1/\delta_{er} \\ n &= 1 \\ n' &= -1/L_n \\ B &= 1 \\ T_e &= 1 \\ L_n &= (RB_p n T_e^{3/2} / B) / (.51\nu_r / \delta_{er}) \\ k_{\parallel} &= \omega_s \sqrt{\beta_r \sigma_{\parallel}} \\ \omega_s &= k_b RB_p n / BL_n \end{aligned}$$

The input data for these test cases is also shown in table 1.

The results of this are shown in Fig. 1-2. The data for these plots is additionally shown in tables 2-3. In these figures, the upper curves are growth rates while the lower curves are frequencies. The lines are analytic solutions, and the dots are numerical solutions.

The two sets of dots arise because the analytic solution assumes that the parallel wavenumber is equal to the smallest nonzero wavenumber possible for a given size domain. This is not true for large domain sizes, since in that case the fastest growing mode has a wavelength shorter than the domain size. For this reason, if the eigenvalue solver returns the fastest growing mode it will not agree with the analytic solution for low σ_{\parallel} . The solution to this is to return a number of fast-growing eigenmodes and sort them by parallel wavenumber, which can be calculated from the eigenvector. This returns the growth rate of the fundamental (longest wavelength) mode, which agrees with the analytic solution.

References

- [1] W. Horton, Rev. Mod. Phys. **71**, 735 (1999).
- [2] M. Wakatani and A. Hasegawa, Phys. Fluids **27**, 611 (1984).

nx	2	ny	64
dx	100000	dy	.0997331
ω_{ci}	39.2156862745098	n	0
μ	12.5	ν_r	0.05
q	0	k_b	.5
j	$.00360624\sqrt{\sigma_{\parallel}}$	k_{ψ}	0
κ_n	0	κ_g	0
B	1	RB_p	1
n_0	1	T_e	1
n'_0	-.051	δ_{er}^2	4

Table 1: Non-dimensional profile functions and parameters used in the resistive drift wave test case, as a function of the parameter σ_{\parallel} .

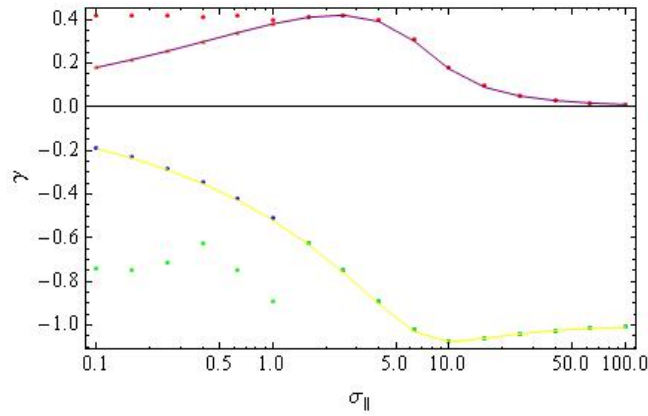


Figure 1: Growth rate vs. parallel scale for electrostatic RDW model. Upper curves are growth rates, lower curves are frequencies. Yellow/purple curves are analytic solutions, orange/blue points are 2DX results for the fundamental mode, red/green points are 2DX results for the dominant mode.

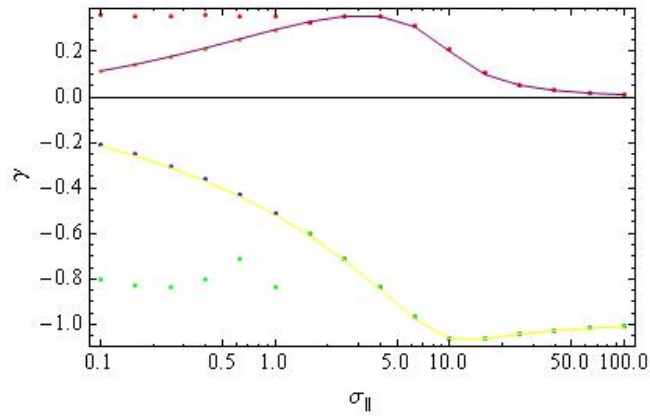


Figure 2: Growth rate vs. parallel scale for electromagnetic RDW model. Upper curves are growth rates, lower curves are frequencies. Yellow/purple curves are analytic solutions, orange/blue points are 2DX results for the fundamental mode, red/green points are 2DX results for the dominant mode.

$\sigma_{\parallel}/\sigma_{\perp}$	γ (fundamental)	γ (dominant)	γ (analytic)
.1	.178224	.420647	.180541
.158489	.213252	.420558	.215866
.251189	.252259	.420242	.255123
.398107	.294227	.408777	.297238
.630957	.337129	.420527	.340093
1	.377343	.395195	.379923
1.58489	.408656	.408656	.410281
2.51189	.420541	.420542	.420245
3.98107	.395456	.395457	.391685
6.30957	.310130	.310130	.301662
10	.181312	.181313	.173175
15.8489	.093838	.0938385	.0897538
25.1189	.051386	.0513868	.0493784
39.8107	.029746	.0297466	.0286701
63.0957	.017815	.0178157	.0172016
100	.010887	.0108879	.0105238

Table 2: Growth rate vs. parallel scale for electrostatic RDW model.

$\sigma_{\parallel}/\sigma_{\perp}$	γ (fundamental)	γ (dominant)	γ (analytic)
.1	.113059	.359233	.114909
.158489	.14177	.357119	.143972
.251189	.175374	.35643	.1779
.398107	.213144	.359023	.215916
.630957	.253536	.35487	.256408
1	.293914	.35626	.296647
1.58489	.330003	.330003	.332215
2.51189	.35479	.35479	.355837
3.98107	.356342	.356342	.355114
6.30957	.314289	.314289	.309006
10	.209467	.209467	.200683
15.8489	.105075	.105075	.100045
25.1189	.0545967	.0545967	.052327
39.8107	.0307527	.0307527	.029601
63.0957	.0181601	.0181601	.0175218
100	.0110127	.0110127	.0106402

Table 3: Growth rate vs. parallel scale for electromagnetic RDW model.

Ion temperature gradient mode test of the 2DX code

D. A. Baver and J. R. Myra

Lodestar Research Corporation, Boulder Colorado 80301

Maxim Umansky

Lawrence Livermore National Laboratory

1 Introduction

This test was devised to verify the ability of the 2DX eigenvalue code to correctly solve a simple fluid model relevant to edge turbulence in tokamaks. Since the functionality of the 2DX code depends on both the source code itself and the input file defining the system of equations to solve (structure file), this test demonstrates both. Since a similar test was performed on an earlier version of 2DX, this verifies that the current version retains this functionality. Moreover, since the structure file for this test represents a subset of a more general 6-field model, many of the terms in that test are also verified.

This test compares 2DX results to BOUT simulations, and to exact analytic results based on a simplified eigenvalue problem.

2 Description

2.1 Code structure

The 2DX code is a highly flexible eigenvalue solver designed for problems relevant to edge physics in toroidal plasma devices. Its flexibility stems from the use of a specialized input file containing instructions on how to set up a particular set of equations. Because of this, the 2DX code permits model equations to be changed without altering its source code. The drawback to this approach is that any change to the structure file represents a potential source of error, necessitating re-verification. This problem is offset by the fact that the source code remains unchanged, thus testing one structure file builds confidence in the underlying code that interprets the structure file. Also, structure files can be translated into analytic form, thus allowing the user to verify that the file contains the equations intended.

The structure file contains two main parts: an elements section, which constructs the differential operators and other functions used in a particular set of equations, and a formula section, which assembles these into an actual set of equations. This separation means that elements can be recycled in other structure files. By testing one structure file, one builds confidence in the elements used in that file. The main source of error when switching to a different structure file then is in the formula section, which can be manually verified by translating into analytic form.

Regardless of the content of the structure file, the 2DX code is fundamentally a finite-difference eigenvalue solver. As such, it is subject to the limitations of any code of its type.

2.2 Model equations

For this test we use the following model equations [1]-[2]:

$$\gamma \nabla_{\perp}^2 \delta \Phi = -\frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (1)$$

$$\gamma \delta n = -n \partial_{\parallel} \delta u - \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (2)$$

$$\gamma \delta u = -\frac{1}{n} \nabla_{\parallel} \delta p \quad (3)$$

$$\gamma \delta T_i = -\delta v_E \times \nabla T_i - \frac{2}{3} T_i \partial_{\parallel} \delta u \quad (4)$$

$$-\gamma \nabla_{\perp}^2 \delta A = -\mu n \nabla_{\parallel} \delta \Phi + \mu T_e \nabla_{\parallel} \delta n \quad (5)$$

where

$$\delta p = (T_e + T_i) \delta n + n (\delta T_e + \delta T_i) \quad (6)$$

$$C_r = \mathbf{b} \times \kappa \cdot \nabla = -\kappa_g R B_p \partial_x + i(\kappa_n k_b - \kappa_g k_{\psi}) \quad (7)$$

$$\nabla_{\perp}^2 = -k_b^2 - jB(k_{\psi} - i\partial_x R B_p)(1/jB)(k_{\psi} - iR B_p \partial_x) \quad (8)$$

$$\partial_{\parallel} Q = B \nabla_{\parallel} (Q/B) \quad (9)$$

$$\nabla_{\parallel} = j \partial_y \quad (10)$$

$$\delta v_E \cdot \nabla Q = -i \frac{k_z (R B_p \partial_x Q)}{B} \delta \Phi \quad (11)$$

$$\nu_e = .51 \nu_r n / T_e^{3/2} \quad (12)$$

2.3 Boundary conditions

This test case uses phase-shift periodic boundary conditions in the parallel direction, and zero-derivative boundary conditions in the radial direction. The phase shift in the parallel direction is given by:

$$e^{i2\pi nq} \quad (13)$$

2.4 Profile setup

The formulas in Eq. 1-5 are normalized to Bohm units. Values are converted by dividing input distances by ρ_s , and input magnetic fields are in Tesla. Output eigenvalues are multiplied by ω_{ci} . Resistivity is given by the formula:

$$\nu_r = \frac{\mu_u}{.51\sigma} \quad (14)$$

where

$$\sigma = 1.96 \frac{\omega_{ce}}{\nu_{ei}} \quad (15)$$

The geometry used is a periodic slab. Curvature effects are included in the equation set, but curvature is set to zero. Zero-derivative boundary conditions are used in the radial direction, and the domain is set to only two grid cells wide in that direction. This is done so as to approximate a 1-D problem using a 2-D code, and because the 2DX code cannot simulate domains that are only one grid cell wide in either direction.

3 Analytic results

Since the ITG model is tested in a homogenous domain, it can be solved analytically by taking a Fourier transform in both directions. This allows the system of differential equations to be reduced to algebraic matrix equations. Assuming that $\nabla_{\perp}^2 = -k_b^2$, this yields an eigenvalue problem of the form $Ax = \gamma x$, where A is:

$$\begin{bmatrix} 0 & -ik_y & 0 & 0 & ik_y \\ -ik_y(T_e + T_i)/n & 0 & -ik_y & 0 & 0 \\ ik_b T_i'/n & -(2/3)iT_i k_y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -ik_y/nk_b^2 \\ i\mu n k_y & 0 & 0 & -i\mu n k_y & 0 \end{bmatrix} \quad (16)$$

and x is $[\delta\Phi, \delta n, \delta u, \delta T_i, \delta A]$.

These can be solved using standard eigenvalue solving routines. The results of this calculation are shown in Figs. 1 along with the numerical results from 2DX.

4 Numerical results

The code was tested by running one case and sorting the eigenvalues by parallel wavenumber. This allowed for a plot of growth rate as a function of wavenumber using only a single run of the code. The parameters used in this run are as follows:

$$\begin{aligned} n &= 1 \\ T_e &= 1 \\ T_i &= 1 \\ T'_i &= -1 \\ B &= 1 \\ RB_p &= 1 \\ k_b &= 1 \\ j &= .07 \\ \omega_{ci} &= 1 \end{aligned}$$

The profiles used in this test case are also shown in table 1.

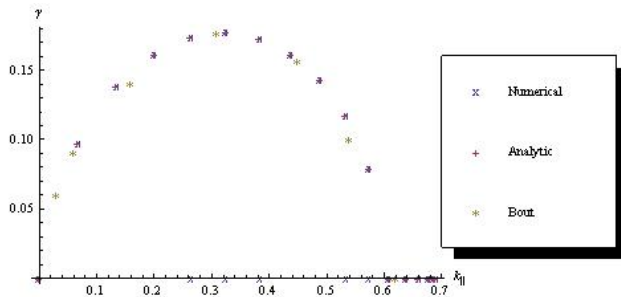
The results are shown in Fig. 1, and the raw data for this run is shown in table 2. The vertical axis here is normalized growth rate, whereas the horizontal axis is normalized parallel wavenumber. For analytic and BOUT results, the parallel wavenumber is specified in order to calculate each specific data point. For the 2DX results, parallel wavenumber is not specified, and the eigenvalue solver returns a large number (in this case 60) of relevant eigenmodes. From the eigenvector of each eigenmode one can calculate a parallel wavenumber, and that value is used to determine the horizontal position of each data point. In this graph, there are fewer than 60 unstable modes in the system, so a number of modes on the neutral branch of the dispersion relation are returned as well.

References

- [1] B. Coppi, M.N. Rosenbluth and R.Z. Sagdeev, Phys. Fluids **10**, 582 (1967).
- [2] S. Hamaguchi and W. Horton, Phys. Fluids B **2**, 1833 (1990).

nx	4	ny	16
dx	.693242	dy	.392699
ω_{ci}	4.79×10^7	m	.5
nx_{LCS}	4	nx_{mis}	0
j_1	1	j_2	16
Γ	0	μ_{ii}	0
μ	2	ν_r	.00131267
δ_{er}^2	1	S_E	1
Λ_1	1	Λ_2	0
q	1.5	k_b	-.144249

Table 1: Profile functions and parameters used in the ITG test case.

Figure 1: Growth rate vs. k_y for the ITG model

k_z	γ	k_z	γ	k_z	γ
0.325609	0.177017	0.325609	0.177017	0.325609	0.177017
0.325609	0.177017	0.264332	0.173671	0.264332	0.173671
0.264332	0.173671	0.264332	0.173671	0.38375	0.172702
0.38375	0.172702	0.38375	0.172702	0.38375	0.172702
0.200509	0.161416	0.200509	0.161416	0.200509	0.161416
0.200509	0.161416	0.438196	0.161362	0.438196	0.161362
0.438196	0.161362	0.438196	0.161362	0.488422	0.143084
0.488422	0.143084	0.488422	0.143084	0.488422	0.143084
0.134755	0.137856	0.134755	0.137856	0.134755	0.137856
0.134755	0.137856	0.533943	0.117025	0.533943	0.117025
0.533943	0.117025	0.533943	0.117025	0.0677036	0.0973153
0.0677036	0.0973153	0.0677036	0.0973153	0.0677036	0.0973153
0.574323	0.0793186	0.574323	0.0793186	0.574323	0.0793186
0.574323	0.0793186	0	0	0	0
0.574323	0	0.574323	0	0.690732	0
0.574323	0	0.533943	0	0.638154	0
0.67746	0	0.609172	0	0.66099	0
0.533943	0	0.533943	0	0.325609	0
0.264332	0	0.38375	0	0.687406	0
0.67746	0	0.325609	0	0.66099	0

Table 2: Growth rate vs. k_y for the ITG model

Geodesic acoustic mode test of the 2DX code

D. A. Baver and J. R. Myra

Lodestar Research Corporation, Boulder Colorado 80301

Maxim Umansky

Lawrence Livermore National Laboratory

1 Introduction

This test was devised to verify the ability of the 2DX eigenvalue code to correctly solve a simple fluid model relevant to edge turbulence in tokamaks. Since the functionality of the 2DX code depends on both the source code itself and the input file defining the system of equations to solve (structure file), this test demonstrates both. Since a similar test was performed on an earlier version of 2DX, this verifies that the current version retains this functionality. Moreover, since the structure file for this test represents a subset of a more general 6-field model, many of the terms in that test are also verified.

This test compares 2DX results to approximate analytic solutions for relevant eigenmodes of interest.

2 Description

2.1 Code structure

The 2DX code is a highly flexible eigenvalue solver designed for problems relevant to edge physics in toroidal plasma devices. Its flexibility stems from the use of a specialized input file containing instructions on how to set up a particular set of equations. Because of this, the 2DX code permits model equations to be changed without altering its source code. The drawback to this approach is that any change to the structure file represents a potential source of error, necessitating re-verification. This problem is offset by the fact that the source code remains unchanged, thus testing one structure file builds confidence in the underlying code that interprets the structure file. Also, structure files can be translated into analytic form, thus allowing the user to verify that the file contains the equations intended.

The structure file contains two main parts: an elements section, which constructs the differential operators and other functions used in a particular set

of equations, and a formula section, which assembles these into an actual set of equations. This separation means that elements can be recycled in other structure files. By testing one structure file, one builds confidence in the elements used in that file. The main source of error when switching to a different structure file then is in the formula section, which can be manually verified by translating into analytic form.

Regardless of the content of the structure file, the 2DX code is fundamentally a finite-difference eigenvalue solver. As such, it is subject to the limitations of any code of its type.

2.2 Model equations

For this test we use the following model equations [1]:

$$\gamma \nabla_{\perp}^2 \delta \Phi = + \frac{2B}{n} C_r \delta p - \frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \delta A + \Gamma \nabla_{\perp}^2 \delta \Phi + \mu_{ii} \nabla_{\perp}^4 \delta \Phi \quad (1)$$

$$\gamma \delta n = \frac{2}{B} (C_r \delta p_e - n C_r \delta \Phi) - n \partial_{\parallel} \delta u - \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (2)$$

$$\gamma \delta u = - \frac{1}{n} \nabla_{\parallel} \delta p \quad (3)$$

$$- \gamma \nabla_{\perp}^2 \delta A = \nu_e \nabla_{\perp}^2 \delta A - \mu n \nabla_{\parallel} \delta \Phi + \mu T_e \nabla_{\parallel} \delta n \quad (4)$$

$$\delta p = (T_e + T_i) \delta n + n (\delta T_e + \delta T_i) \quad (5)$$

$$C_r = \mathbf{b} \times \kappa \cdot \nabla = -\kappa_g R B_p \partial_x + i(\kappa_n k_b - \kappa_g k_{\psi}) \quad (6)$$

$$\nabla_{\perp}^2 = -k_b^2 - jB(k_{\psi} - i\partial_x R B_p)(1/jB)(k_{\psi} - iR B_p \partial_x) \quad (7)$$

$$\partial_{\parallel} Q = B \nabla_{\parallel} (Q/B) \quad (8)$$

$$\nabla_{\parallel} = j \partial_y \quad (9)$$

$$\delta v_E \cdot \nabla Q = -i \frac{k_z (R B_p \partial_x Q)}{B} \delta \Phi \quad (10)$$

$$\nu_e = .51 \nu_r n / T_e^{3/2} \quad (11)$$

2.3 Boundary conditions

This test case uses phase-shift periodic boundary conditions in the parallel direction, and zero-derivative boundary conditions in the radial direction. The phase shift in the parallel direction is given by:

$$e^{i2\pi n q} \quad (12)$$

2.4 Profile setup

The formulas in Eq. 1-4 are normalized to Bohm units. Values are converted by dividing input distances by ρ_s , and input magnetic fields are in Tesla. Output eigenvalues are multiplied by ω_{ci} . Resistivity is given by the formula:

$$\nu_r = \frac{\mu}{.51\sigma} \quad (13)$$

where

$$\sigma = 1.96 \frac{\omega_{ce}}{\nu_{ei}} \quad (14)$$

The geometry used is a thin toroidal annulus with major radius R . Other geometric effects are neglected, so that the domain is effectively a shearless slab. Both normal and geodesic curvature are calculated as follows:

$$\kappa_n = \frac{\cos(y)}{R} \quad (15)$$

$$\kappa_g = \frac{\sin(y)}{R} \quad (16)$$

Parallel derivatives are calculated using the Jacobian factor $1/qR$. Toroidal mode number is set to zero. Binormal wavenumber is therefore also equal to zero. In addition, a constant parameter Γ is used to give the GAM a positive growth rate; physically, this represents nonlinear drive of the GAM by turbulence.

3 Analytic results

Since the eigenmodes of the GAM model are not homogenous, there is no exact analytic solution. An approximate solution can be found by assuming that δA terms are small and that geodesic curvature has the form $\kappa_g = \hat{\kappa}_g \sin \theta \mathbf{e}_\theta$. For purposes of this calculation, we normalize n and T to reference values, set μ_{ii} to zero, and set the Jacobian factor $j = 1/qR$. This results in the following dispersion relation:

$$i\omega(\omega^2 - \omega_g^2) = -\Gamma(\omega^2 - \omega_s^2) \quad (17)$$

where

$$\omega_s = k_{\parallel} c_s \quad (18)$$

$$\omega_g^2 = \frac{c_s^2}{R^2} \left(2 + \frac{1}{q^2} \right) \quad (19)$$

If Γ is small, this yields a frequency and growth rate:

$$\omega = \omega_g + \frac{i\Gamma}{2 + 1/q^2} \quad (20)$$

In addition, there is also a zero frequency mode which can be identified as a zonal flow. This has a growth rate:

$$\omega = \frac{i\Gamma}{1 + 2q^2} \quad (21)$$

These approximations break down at certain specific values of q . This occurs because of a resonance between the GAM and spatial harmonics of the sound wave. The sound wave has frequency:

$$\omega_s = \frac{m^2}{q^2 R^2} \quad (22)$$

Consequently, a resonance between ω_s and ω_g occurs at values of q that satisfy the equation:

$$q = \sqrt{\frac{m^2 - 1}{2}} \quad (23)$$

4 Numerical results

The code was tested by sweeping the variable q and plotting the fastest growing eigenvalue. The parameters used in this test are as follows:

$\rho_s = 1$
 $R = 1000$
 $\delta a = 100$
 $n = 1$
 $T_e = 1$
 $T_i = 0$
 $\delta_{er} = 1$
 $\mu = 1836$
 $k_\psi = .2$
 $\Gamma = .00001$
 $B = 1$
 $RB_p = 1$
 $\omega_{ci} = 1$

A more complete list of input values is shown in table 1.

The results of this test are shown in Fig. 1. The raw data is also shown in table 2. The red line is the growth rate predicted for the zonal flow mode, whereas the yellow line is the growth rate predicted for the analytic GAM. The blue line is the 2DX result. Note that the 2DX result agrees with whichever of the two analytic results is greater, except at certain specific q values. These values agree with those predicted by the formula in Eq. 23.

References

- [1] N. Winsor, J.L. Johnson and J.M. Dawson, Phys. Fluids **11**, 2448 (1968).

nx	4	ny	32
dx	33.3333	dy	.202683
ω_{ci}	1	n	0
μ	1836	ν_r	.0000196078
q	q_0	k_b	0
j	$.001/q_0$	k_ψ	.2
κ_n	$.001 \cos(y)$	κ_g	$.001 \sin(y)$
B	1	RB_p	1
n_0	1	T_e	1
Γ	.00001		

Table 1: Non-dimensional profile functions and parameters used in the GAM test case, as a function of q_0 .

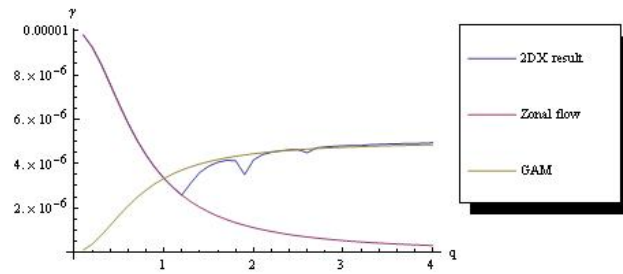


Figure 1: Growth rate vs. q for the GAM model.

q	$\gamma \times 10^6$	Zonal flow	Analytic GAM	q	$\gamma \times 10^6$	Zonal flow	Analytic GAM
.1	9.80481	9.80392	.098039	2.1	4.41140	1.01833	4.49084
.2	9.26424	9.25926	.370370	2.2	4.51496	.936330	4.53184
.3	8.48054	8.47458	.762712	2.3	4.57988	.863558	4.56822
.4	7.58423	7.57576	1.21212	2.4	4.62225	.798722	4.60064
.5	6.67693	6.66667	1.66667	2.5	4.62246	.740741	4.62963
.6	5.82520	5.81395	2.09302	2.6	4.49110	.688705	4.65565
.7	5.06206	5.05051	2.47475	2.7	4.70994	.641849	4.67908
.8	4.39735	4.38596	2.80702	2.8	4.75274	.599520	4.70024
.9	3.82771	3.81579	3.09160	2.9	4.78082	.561167	4.71942
1.0	3.34361	3.33333	3.33333	3.0	4.80332	.526316	4.73684
1.1	2.93355	2.92398	3.53801	3.1	4.81994	.494560	4.75272
1.2	2.58617	2.57732	3.71134	3.2	4.81926	.465549	4.76723
1.3	3.12899	2.28311	3.85845	3.3	4.85264	.438982	4.78051
1.4	3.60623	2.03252	3.98374	3.4	4.86981	.414594	4.79270
1.5	3.88574	1.81818	4.09091	3.5	4.88342	.392157	4.80392
1.6	4.06130	1.63399	4.18301	3.6	4.89546	.371471	4.81426
1.7	4.15147	1.47493	4.26254	3.7	4.90607	.352361	4.82382
1.8	4.13123	1.33690	4.33155	3.8	4.91024	.334672	4.83266
1.9	3.50439	1.21655	4.39173	3.9	4.92539	.318269	4.84087
2.0	4.16704	1.11111	4.44444	4.0	4.93414	.303030	4.84848

Table 2: Growth rate vs. q for the GAM model.

Ideal kink mode benchmark

J. R. Myra, D. A. Baver

Lodestar Research Corp., 2400 Central Ave. P-5, Boulder, Colorado 80301

M. V. Umansky

Lawrence Livermore National Laboratory, Livermore, CA 94550

A. Introduction

This test was devised to verify the ability of the 2DX eigenvalue code to correctly solve a fluid model relevant to Edge Localized Mode (ELM) physics in tokamaks, viz. the current-driven ideal kink mode. Kink/peeling mode instability physics, in combination with pressure gradient-driven instability physics is believed to be responsible for Type I ELM onset.¹ Since the functionality of the 2DX code depends on both the source code itself and the input file defining the system of equations to solve (structure file), this test demonstrates both. Similar tests have been performed using other physics models. Moreover, since the structure file for these tests represents a subset of a more general 6-field model, many of the terms in that test are also verified. A more detailed description of the 2DX code can be found in Ref. 2.

The present test compares 2DX results to asymptotic analytic results based on a sharp boundary solution³ of the eigenvalue problem, with some additional analytic embellishments to Ref. 3 described here. In particular, a strict sharp boundary limit is unsuitable as a benchmark test case because it cannot be resolved numerically with a finite grid. Thus, we extend the analytical result to account for small departures from the strict sharp boundary limit.

B. Ideal kink mode model

To benchmark the current gradient drive term in the ideal MHD model, we solve the equation

$$\gamma^2 \left(\nabla_{\perp}^2 \delta\Phi \right) = \mu \delta_{er}^2 \frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \nabla_{\parallel} \delta\Phi - \frac{ik_b B (\partial_r J_{\parallel})}{n} \nabla_{\parallel} \delta\Phi \quad (1)$$

This equation may be obtained from the primitive six-field eigenvalue equations (see e.g. Ref. 2) by combining the equations for vorticity and Ohm-Ampere's law in the ideal MHD limit. Here, we work in Bohm-normalized variables with times normalized to $1/\Omega_i$, lengths normalized to a reference sound gyroradius ρ_{sr} , temperature and electrostatic potential Φ/e to a reference value of electron temperature T_{er} , and density to a reference value of density n_{er} . In Eq. (1), $\mu = m_i/m_e$, $\delta_{er}^2 = c^2/(\rho_{sr}^2 \omega_{pe}^2)$ and for any Q ,

$\partial_{\parallel} Q = B \nabla_{\parallel} (B^{-1} Q)$, $\partial_r = R B_p \partial / \partial \psi$, k_b is the binormal component of the perpendicular wavenumber; other symbols have their usual meanings.

For the benchmark test set we take $B = n = 1$, and since there is no variation in the equilibrium along B , for analytical work $\nabla_{\parallel} = i k_{\parallel}$ (in the code, parallel variation is solved for numerically).

$$\left(\gamma^2 + \mu \delta_{er}^2 k_{\parallel}^2 \right) \left(\nabla_{\perp}^2 \delta \Phi \right) = k_b k_{\parallel} (\partial_r J_{\parallel}) \delta \Phi \quad (2)$$

The current profile is taken as a sharply varying step function in the next section, while corrections for smoothed profiles are given in the Appendix. The analytical solution and benchmark test is carried out in the cylindrical tokamak model.

B. Analytical solution

For a strict sharp boundary J_{\parallel} profile, ($J_{\parallel} = J_{\parallel 0}$ to the left, and 0 to the right) with step at $r = a$, we solve the equation

$$\nabla_{\perp}^2 \delta \Phi = (\partial_r^2 - k_b^2) \delta \Phi = 0 \quad (3)$$

in each region to obtain

$$\delta \Phi = \exp(\pm k_b (r - a)) \quad (4)$$

Then integrate Eq. (2) across the step to get the jump condition

$$\left(\gamma^2 + \mu \delta_{er}^2 k_{\parallel}^2 \right) \left(\partial_r \delta \Phi \right)_{r=a} = -k_b k_{\parallel} J_{\parallel 0} \delta \Phi \quad (5)$$

where $[J_{\parallel}] = -J_{\parallel 0}$. The dispersion relation is obtained by employing Eq. (4) in Eq. (5)

$$\gamma^2 + \mu \delta_{er}^2 k_{\parallel}^2 - k_{\parallel} J_{\parallel 0} / 2 = 0 \quad (6)$$

In dimensional units

$$\omega^2 - k_{\parallel}^2 v_a^2 + k_{\parallel} \frac{J_{\parallel 0}}{2} = 0 \quad (7)$$

where v_a is the Alfvén velocity. For the sharp boundary equilibrium model (making J_{\parallel} consistent with q)

$$J_{\parallel 0} = \frac{2v_a^2}{qR} \rightarrow \frac{2\mu \delta_{er}^2 \rho_s}{qR} \quad (8)$$

$$\hat{\omega}^2 - \hat{k}_{\parallel}^2 + \hat{k}_{\parallel} = 0 \quad (9)$$

where $\hat{\omega} = \omega q R / v_a$, $\hat{k}_{\parallel} = k_{\parallel} q R$. Maximum growth is at $\hat{k}_{\parallel} = 1/2$ and is $\hat{\omega} = i/2$. This implies

$$\gamma_{\max} = \frac{\mu^{1/2} \delta_{\text{er}}}{2qR} \quad (\text{Bohm dimensionless})$$

This result is similar to that obtained in Ref. 3, but the inertial term is twice as large here because we have assumed a constant density profile (rather than taking it as a step function also.)

The above can be generalized to include a finite wall position, and to correct for a sharp but finite-width tanh function instead of a strict step function. The generalized dispersion relation takes the form

$$\hat{\omega}^2 - \hat{k}_{\parallel}^2 + W\hat{k}_{\parallel} = 0 \quad (10)$$

where W is derived in the Appendix.

C. Numerical solution and comparison

We solve Eq. (1) with 2DX on a $n_x \times n_y = 511 \times 128$ grid. Dimensionless input parameters are $\mu = 3672$, $\delta_{\text{er}}^2 = 0.08449$, $1/(qR) = 1.8758 \times 10^{-4}$ and the reference values employed for dimensional results are $\Omega_i = 9.58 \times 10^7/\text{s}$, $\omega_a = v_a/(qR) = 3.1657 \times 10^5/\text{s}$. The current profile is taken as

$$J_{\parallel} = \frac{J_{\parallel 0}}{2} \left(1 - \tanh \left(\frac{r-a}{z_w} \right) \right) \quad (11)$$

where $z_w = 0.002$ and $J_{\parallel 0}$ is given by Eq. (8).

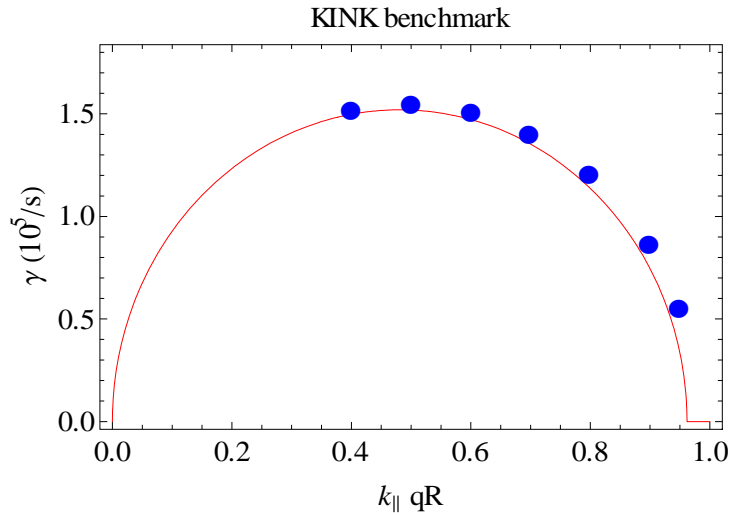


Fig. 1 Growth rate $\gamma(10^5/\text{s})$ vs. normalized k_{\parallel} for 2DX (blue dots) and for the solution of Eq. (10) retaining finite step width corrections for the current gradient. Finite wall position corrections are negligible.

Because of the high resolution, these runs take several hours each, although the resolution of the rather sharp current gradient is still not superb. For these runs, $W = 0.962$ from the finite width corrections, while the finite wall positions are negligible (hence the value of k_b or toroidal mode number $n = 20$ that was employed drops out.) Results are shown in Fig. 1.

In Fig. 1, cases with $k_{\parallel}qR < 0.5$ are awkward numerically since the fundamental parallel mode in the box is not usually the fastest growing mode. In these cases 2DX will report the fastest mode to be a poorly resolved harmonic. The desired mode can be found by picking through the spectrum, but this has not been done except for the case with $k_{\parallel}qR < 0.4$. There is no reason to suspect problems in benchmarking these cases, it is just inconvenient.

Appendix: Corrections for finite wall position and finite width step

For conducting wall BCs, ($\delta\Phi = 0$) at finite locations on each side of the step, $z = \pm z_w$, where $z = r - a$ is a shifted radial variable centered at the step, we can modify Eqs. (3) and (4) as follows

$$\delta\Phi = \exp(-k_b z) \rightarrow \frac{\sinh(k_b(z_w - z))}{\sinh(k_b z_w)} \quad (\text{A1})$$

$$\partial_r \delta\Phi|_{r=a} = -k_b \rightarrow -k_b \coth k_b z_w \quad (\text{A2})$$

So in the dispersion relation, Eq. (7), after dividing through, the drive term $J_{\parallel 0}$ is multiplied by $\tanh(k_b z_w)$. Similarly, one can derive the correction for insulating BCs, $\partial_r \delta\Phi(z_w) = 0$. In summary we find

$$W = \begin{cases} \tanh(k_b z_w), & \text{conducting BCs} \\ \coth(k_b z_w), & \text{insulating BCs} \end{cases} \quad (\text{A3})$$

We can also derive corrections for a finite current profile width. The idea is to do a perturbative expansion in the width z_m where the current profile is given by

$$J(z) = \frac{J_{\parallel 0}}{2} (1 - \tanh(z/z_m)) \quad (\text{A4})$$

In leading order, we take the solution to be that of the step-function model (for simplicity, the infinite wall case), given by Eq. (4). Then in the vicinity of z_m for $k_b z_m \ll 1$ we can take $\delta\Phi = 1$ and an equation accurate through next order is

$$\gamma_a^2 (\partial_z^2 - k_b^2) \delta\Phi = k_b k_{\parallel} (\partial_r J_{\parallel}) \quad (\text{A5})$$

where

$$\gamma_a^2 \equiv \gamma^2 + \mu \delta_{\text{er}}^2 k_{\parallel}^2 \quad (\text{A6})$$

Integrating from $z = 0$ (noting even parity of $\delta\Phi$), and dropping $k_b z_m$ contributions gives

$$\partial_z \delta\Phi = \frac{k_b k_{\parallel}}{\gamma_a^2} (J_{\parallel} - J_{\parallel 0} / 2) \quad (\text{A7})$$

$$\delta\Phi = 1 + \frac{k_b k_{\parallel}}{\gamma_a^2} \int_0^z dz J_{\parallel} - \frac{k_b k_{\parallel}}{2\gamma_a^2} J_{\parallel 0} z \quad (\text{A8})$$

Next we match the logarithmic derivatives at a point z_1 such that

$$z_m \ll z_1 \ll 1/k_b \quad (\text{A9})$$

i.e. z_1 is outside the current profile gradient, but inside the region where $k_b z_m \ll 1$ holds. The matching condition is

$$\left(\frac{\partial_z \delta\Phi}{\delta\Phi} \right)_{z_1} = -k_b \quad (\text{A10})$$

$$\frac{-\frac{k_b k_{\parallel} J_{\parallel 0}}{2\gamma_a^2}}{1 + \frac{k_b k_{\parallel}}{\gamma_a^2} \int_0^{z_1} dz J_{\parallel} - \frac{k_b k_{\parallel}}{2\gamma_a^2} J_{\parallel 0} z_1} = -k_b \quad (\text{A11})$$

After cleaning up, this reduces to

$$\gamma^2 + \mu \delta_{\text{er}}^2 k_{\parallel}^2 - \frac{k_{\parallel} J_{\parallel 0}}{2} \left(1 - 2k_b \int_0^{z_1} dz \frac{J_{\parallel}}{J_{\parallel 0}} \right) = 0 \quad (\text{A12})$$

where the $J_{\parallel 0} z_1$ term in Eq. (11) was dropped since it is small in $k_b z_1$ relative to the other $J_{\parallel 0}$ term. This correction enters in the form shown in Eq. (10) and gives

$$W = 1 - 2k_b \int_0^{\infty} dz \frac{1}{2} (1 - \tanh(z/z_m)) \quad (\text{A13})$$

or

$$W = 1 - k_b z_m \ln 2 \quad (\text{A14})$$

If both finite wall and finite width corrections need to be applied, and they are both small corrections, the W 's can be multiplied together.

References

1. P. B. Snyder, H. R. Wilson, and X. Q. Xu, Phys. Plasmas **12**, 056115 (2005).
2. D. A. Baver, J. R. Myra and M.V. Umansky, Comp. Phys. Comm. **182**, 1610, (2011).
3. D. A. D'Ippolito, J. R. Myra, S.C. Jardin, M.S. Chance and E.J. Valeo, Phys. of Plasmas **2**, 3429 (1995); and refs therein.

Parallel Kelvin-Helmholtz mode benchmark

J. R. Myra, D. A. Baver

Lodestar Research Corp., 2400 Central Ave. P-5, Boulder, Colorado 80301

M. V. Umansky

Lawrence Livermore National Laboratory, Livermore, CA 94550

A. Introduction

This test was devised to verify the ability of the 2DX eigenvalue code to correctly solve a fluid model relevant to edge turbulence in tokamaks, viz. the parallel Kelvin-Helmholtz mode.^{1,2} Since the functionality of the 2DX code depends on both the source code itself and the input file defining the system of equations to solve (structure file), this test demonstrates both. Similar tests have been performed using other physics models. Moreover, since the structure file for these tests represents a subset of a more general 6-field model, many of the terms in that test are also verified. A more detailed description of the 2DX code can be found in Ref. 3.

The present test compares 2DX results to exact semi-analytic results based on a local limit of the eigenvalue problem. The semi-analytic results are equivalent to the numerical solution of a polynomial dispersion relation, i.e. they are obtained without discretization.

B. Parallel Kelvin-Helmholtz 4-field model

A 4-field model containing the physics of the parallel Kelvin-Helmholtz (pKH) mode is^{1,2}

$$\gamma \nabla_{\perp}^2 \delta \Phi + \gamma \frac{T_i}{n} \nabla_{\perp}^2 \delta n = -u \nabla_{\parallel} \nabla_{\perp}^2 \delta \Phi + \frac{B^2}{n} \partial_{\parallel} \delta J + \mu_{ii} \nabla_{\perp}^4 \delta \Phi \quad (1)$$

$$\gamma \delta n = -u \nabla_{\parallel} \delta n - \delta v_E \cdot \nabla n - n \partial_{\parallel} \delta u + \partial_{\parallel} \delta J \quad (2)$$

$$\gamma \delta u = -u \nabla_{\parallel} \delta u - \delta v_E \cdot \nabla u - \frac{T_e + T_i}{n} \nabla_{\parallel} \delta n + \partial_{\parallel} \mu_{\parallel} \nabla_{\parallel} \delta u \quad (3)$$

$$\gamma \delta J = -u \nabla_{\parallel} \delta J - v_e \delta J - \mu n \nabla_{\parallel} \delta \Phi + \mu T_e \nabla_{\parallel} \delta n \quad (4)$$

where we work in Bohm-normalized variables with times normalized to $1/\Omega_i$, lengths normalized to ρ_s , temperature and electrostatic potential Φ/e to a reference value of T_e , and density to a reference value of n_e . Here $\mu = m_i/m_e$, and for any Q , $\partial_{\parallel} Q = B \nabla_{\parallel}$ ($B^{-1}Q$), $\partial_r = RB_p \partial / \partial \psi$, k_b is the binormal component of the perpendicular wavenumber; other symbols have their usual meanings. In particular $\delta \Phi$, δn , δu and δJ

are respectively the perturbations of electrostatic potential, density, parallel velocity and parallel current. See Ref. 3 for a complete description of the six-field model.

In the local limit, and choosing local values for Ω_i , ρ_s , T_e and n_e , we set $n = B = T_e = 1$, $T_i = \tau$, and working in the frame $u = 0$, these equations reduce to

$$\gamma\delta\Phi + \tau\gamma\delta n = -i\frac{k_{\parallel}}{k_{\perp}^2}\delta J - \mu_{ii}k_{\perp}^2\delta\Phi \quad (5)$$

$$\gamma\delta n = ik_b(\partial_r n)\delta\Phi - ik_{\parallel}\delta u + ik_{\parallel}\delta J \quad (6)$$

$$\gamma\delta u = ik_b(\partial_r u)\delta\Phi - (1 + \tau)ik_{\parallel}\delta n - \mu_{\parallel}k_{\parallel}^2\delta u \quad (7)$$

$$\gamma\delta J = -v_e\delta J - \mu_{ik_{\parallel}}\delta\Phi + \mu_{ik_{\parallel}}\delta n \quad (8)$$

We can eliminate the $\gamma\delta n$ term from vorticity to obtain

$$\gamma\delta\Phi = -i\tau k_b(\partial_r n)\delta\Phi + i\tau k_{\parallel}\delta u - i\tau k_{\parallel}\delta J - i\frac{k_{\parallel}}{k_{\perp}^2}\delta J - \mu_{ii}k_{\perp}^2\delta\Phi \quad (9)$$

In Bohm dimensionless units let $\omega_{*e} = k_b(\partial_r n) = k_b/L_n$, $\omega_s = k_{\parallel}$, $\omega_{*u} = k_b/L_u$, $\omega_{*i} = \tau k_b(\partial_r n) = \tau k_b/L_n$. This yields the following set of local equations

$$\gamma\delta\Phi = (-i\omega_{*i} - \mu_{ii}k_{\perp}^2)\delta\Phi + i\tau\omega_s\delta u + \left(-i\tau\omega_s - i\frac{k_{\parallel}}{k_{\perp}^2}\right)\delta J \quad (10)$$

$$\gamma\delta n = i\omega_{*e}\delta\Phi - i\omega_s\delta u + i\omega_s\delta J \quad (11)$$

$$\gamma\delta u = i\omega_{*u}\delta\Phi - i(1 + \tau)\omega_s\delta n - \mu_{\parallel}k_{\parallel}^2\delta u \quad (12)$$

$$\gamma\delta J = -i\mu_{k_{\parallel}}\delta\Phi + i\mu_{k_{\parallel}}\delta n - v_e\delta J \quad (13)$$

We note parenthetically, that the parallel KH mode *exists* in a much simpler 2-field model obtained from the above by dropping the vorticity and δJ equations, assuming Maxwell-Boltzmann electrons, dropping the δJ term in continuity, and the μ_{\parallel} term in the δu equation. This simpler model yields instability for $L_u < L_n$, but has the difficulty that in the plane where both k_b and k_{\parallel} vary to arbitrarily large values, the growth rate increases without bound. Thus the simpler 2-field is neither suitable for benchmarking tests or physics exploration with a discretized code. (If the 2-field model were used in a discretized code, the simulations would be dominated by grid-scale modes regardless of the resolution employed.) The 4-field model with finite μ_{\parallel} solves this problem.

C. Results in the local limit

We choose parameters from the C-Mod QC-mode case, except for L_u the gradient scale length of u , which is varied from strongly unstable ($L_u = 5$) to stable ($L_u = \infty$). The base case parameters in dimensionless Bohm units are $L_n = 12.7$, $\tau = T_i/T_e = 1$, $v_e = 0.34$, $\mu = 3600$, $\mu_{ii} = 0.001$, $\mu_{||} = 2000$. The model is that of Eqs. (10) – (13). The equations were solved using Mathematica to generate local results for comparison with 2DX numerical results.

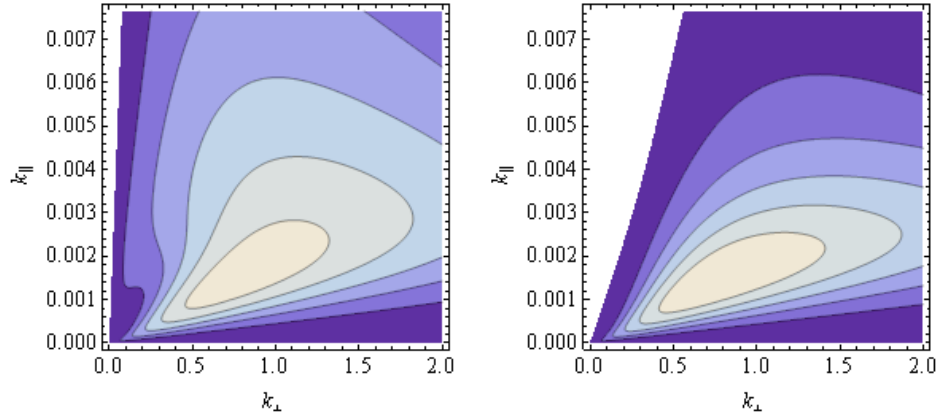


Fig. 1 Growth rate contours for the parallel KH and DW instabilities, for base case parameters (left) and with $1/L_u = 0$ (right). The range of $k_{||}R$ is 40.

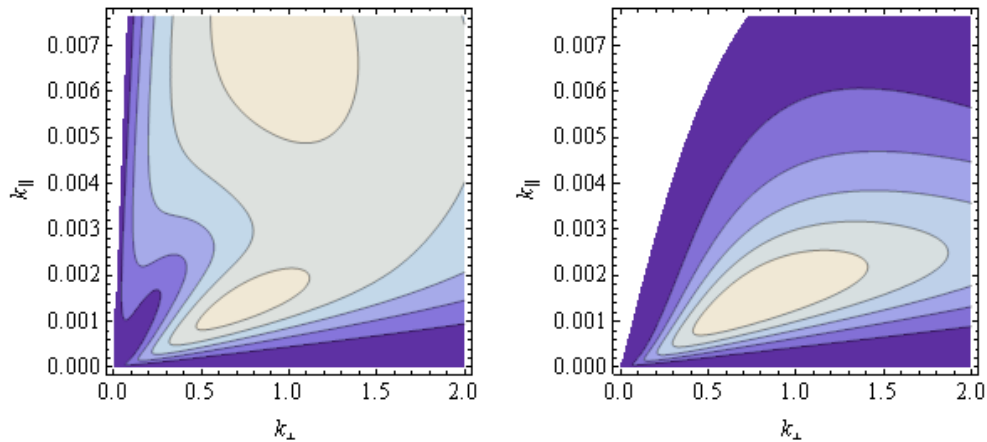


Fig. 2 Same comparison as Fig. 1 except that now $\mu_{||} = 400$. Base case L_u (left) and $1/L_u = 0$ (right).

Results for the base case parameters are shown in Fig. 1. Because of the relatively large (but experimentally realistic) value of μ_{\parallel} , the parallel KH and drift wave (DW) instabilities merge, so the effect of the pKH drive L_u is only to slightly broaden the instability contours.

Fig 2. shows the same comparison for the case $\mu_{\parallel} = 400$. (We cannot take $\mu_{\parallel} = 0$ because then the spectrum never saturates as k_{\parallel} increases.) Now there are two distinct unstable branches: the high k_{\parallel} pKH, and the low k_{\parallel} DW.

D. The parallel KH benchmark test

For a good benchmark test that separates the pKH and DW branches, we choose a case with μ_{\parallel} 5 times smaller than realistic. The parameter set is: (in dimensionless Bohm units) $L_n = 12.7$, $\tau = T_i/T_e = 1$, $v_e = 0.34$, $\mu = 3600$, $\mu_{ii} = 0.001$, $\mu_{\parallel} = 400$. The growth rates in dimensional units for the reference $\Omega_i = 1.982 \times 10^8/\text{s}$ are shown in Fig. 3. The conversion from dimensionless binormal wavenumber k_b to toroidal mode number n is given by $n = 1.04 \times 10^3 k_b$. For this plot, we choose $k_{\parallel \text{base}} = 0.0003$ ($k_{\parallel \text{base}} R = 1.5687$). This is the fundamental mode. Then we calculate growth rates γ for all the m $k_{\parallel \text{base}}$ ($m = 1, 2, 3, \dots$) and pick the maximum γ . This is done for $L_u = 5$ (parallel KH mode) and $L_u = 500$ (remnant DW mode).

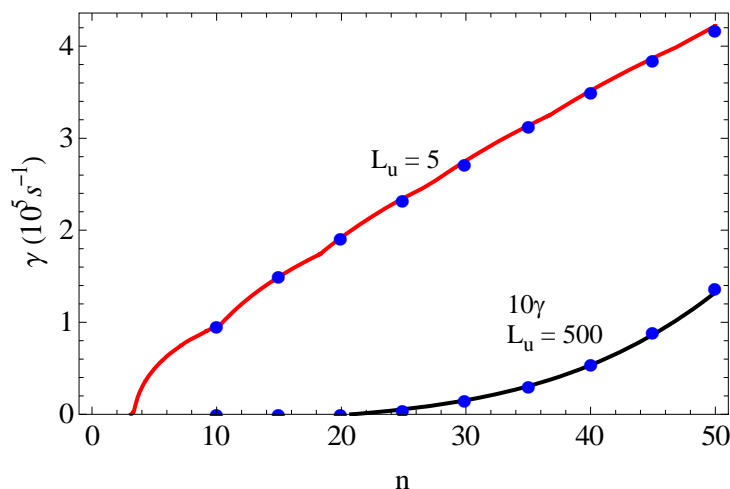


Fig. 3 Semi-analytic growth rates optimized over $m k_{\parallel \text{base}}$ for $L_u = 5$ (red) and 500 (black) (growth rate multiplied by 10). Blue dots are the 2DX results.

n	$\gamma (10^5 \text{ s}^{-1})$	$\gamma (10^5 \text{ s}^{-1})$
	$L_u = 5$	$L_u = 500$
10	0.96151	-0.01152
15	1.50619	-0.00577
20	1.93679	-0.00086
25	2.36788	0.00553
30	2.77891	0.01543
35	3.16148	0.03085
40	3.54409	0.05398
45	3.90071	0.08727
50	4.25273	0.13338

Table 1. Table of semi-analytic growth rates for the benchmark case shown in Fig. 3. These are the target results for the benchmark test of the numerical code (2DX).

References

1. P.J. Catto, M.N. Rosenbluth and C.S. Liu, Phys. Fluids **16**, 1719 (1973).
2. F. Schwander et al., J. Nucl. Mater. in press, (2010), doi:10.1016/j.jnucmat.2010.10.073; and refs. therein
3. D. A. Baver, J. R. Myra and M.V. Umansky, Comp. Phys. Comm. **182**, 1610, (2011).

Toroidal ITG benchmark

J. R. Myra, D. A. Baver

Lodestar Research Corp., 2400 Central Ave. P-5, Boulder, Colorado 80301

M. V. Umansky

Lawrence Livermore National Laboratory, Livermore, CA 94550

A. Introduction

This test was devised to verify the ability of the 2DX eigenvalue code to correctly solve a fluid model relevant to turbulence in tokamaks, viz. the toroidal ion temperature gradient (ITG) mode.^{1,2} Since the functionality of the 2DX code depends on both the source code itself and the input file defining the system of equations to solve (structure file), this test demonstrates both. Similar tests have been performed using other physics models. Moreover, since the structure file for these tests represents a subset of a more general 6-field model, many of the terms in that test are also verified. A more detailed description of the 2DX code can be found in Ref. 3.

The present test compares 2DX results to exact semi-analytic results based on a local limit of the eigenvalue problem. The semi-analytic results are equivalent to the numerical solution of a polynomial dispersion relation, i.e. they are obtained without discretization. The 2DX solutions of the ITG problem given here implement the limit of Maxwell-Boltzmann electrons analytically in the model equations. Although 2DX supports more general electron models, this was done to facilitate comparison with simulation codes using the same approximation.

B. The toroidal ITG 3-field model

From the full 6-fld model described in Ref. 3, one obtains a sublimit appropriate to the toroidal ITG mode^{1,2} as

$$\gamma \nabla_{\perp}^2 \delta \Phi = -i \omega_{*i} \nabla_{\perp}^2 \delta \Phi + \frac{2B}{n} C_r \delta p + \frac{B^2}{n} \partial_{\parallel} \delta J + \mu_{ii} \nabla_{\perp}^4 \delta \Phi \quad (1)$$

$$\gamma \delta n = -\delta v_E \cdot \nabla n + \frac{2}{B} (C_r \delta p_e - n C_r \delta \Phi) + \partial_{\parallel} \delta J \quad (2)$$

$$\gamma \delta T_i = -\delta v_E \cdot \nabla T_i + \frac{2T_i}{3n} \partial_{\parallel} \delta J + \frac{4T_i}{3B} \left(\frac{1}{n} C_r \delta p_e - C_r \delta \Phi - \frac{5}{2} C_r \delta T_i \right) \quad (3)$$

$$\gamma \delta J = -\mu n \nabla_{\parallel} \delta \Phi + \mu T_e \nabla_{\parallel} \delta n \quad (4)$$

where we work in Bohm-normalized variables with times normalized to $1/\Omega_i$, lengths normalized to ρ_s , temperature and electrostatic potential Φ/e to a reference value of T_e , and density to a reference value of n_e . Here $\mu = m_i/m_e$, and for any Q , $\partial_{\parallel} Q = B \nabla_{\parallel} (B^{-1}Q)$, $\partial_r = RB_p \partial / \partial \psi$, k_b is the binormal component of the perpendicular wavenumber; other symbols have their usual meanings.³ In particular $\delta\Phi$, δn , δT_i and δJ are respectively the perturbations of electrostatic potential, density, ion temperature and parallel current.

In the local limit, and choosing local values for Ω_i , ρ_s , T_e and n_e , we set $n = B = T_e = 1$ and $T_i = \tau$. We drop μ_{ij} let $\delta v_E \cdot \nabla n = -ik_b \delta\Phi \partial_r n = i\omega_{*en} \delta\Phi$ where $\omega_{*en} = -k_b \partial_r n$ similarly $\delta v_E \cdot \nabla T_i = i\omega_{*iT} \delta\Phi$ and we note that $\omega_{*iT} = \tau \eta_i \omega_{*en}$. The perturbed pressures in this limit are $\delta p = \delta n(1 + \tau) + \delta T_i$ and $\delta p_e = \delta n$. This gives

$$-\gamma k_{\perp}^2 \delta\Phi = i\omega_{*i} k_{\perp}^2 \delta\Phi + 2C_r [\delta n(1 + \tau) + \delta T_i] + \nabla_{\parallel} \delta J \quad (5)$$

$$\gamma \delta n = -i\omega_{*en} \delta\Phi + 2C_r (\delta n - \delta\Phi) + \nabla_{\parallel} \delta J \quad (6)$$

$$\gamma \delta T_i = -i\omega_{*iT} \delta\Phi + \frac{2\tau}{3} \nabla_{\parallel} \delta J + \frac{4\tau}{3} C_r \left(\delta n - \delta\Phi - \frac{5}{2} \delta T_i \right) \quad (7)$$

$$\gamma \delta J = \mu \nabla_{\parallel} (\delta n - \delta\Phi) \quad (8)$$

where $C_r = -ik_b / R$ and R is the curvature radius. Finally we take the Maxwell-Boltzmann limit analytically to obtain a 3-field model

$$\gamma \nabla_{\perp}^2 \delta\Phi = -i\omega_{*i} \nabla_{\perp}^2 \delta\Phi + 2C_r [\delta\Phi(1 + \tau) + \delta T_i] + \nabla_{\parallel} \delta J \quad (9)$$

$$\gamma \delta\Phi = -i\omega_{*en} \delta\Phi + \nabla_{\parallel} \delta J \quad (10)$$

$$\gamma \delta T_i = -i\omega_{*iT} \delta\Phi + \frac{2\tau}{3} \nabla_{\parallel} \delta J - \frac{10\tau}{3} C_r \delta T_i \quad (11)$$

For the semi-analytical solutions in a slab we also take $\nabla_{\perp}^2 = -k_{\perp}^2$, and $\nabla_{\parallel} = ik_{\parallel}$, while these operators retain their full differential form in the 2DX numerical solution.

If we chose $L_n = 1$, i.e. set $\omega_{*en} = k_b$ then frequencies are effectively renormalized to L_n/c_s as in Sandberg². Furthermore we then identify $C_r = -ik_b \varepsilon_n / 2$, $\omega_{*iT} = \tau \eta_i k_b$, $\omega_{*i} = -\tau(1 + \eta_i) k_b$ [Caution: note the different sign convention in the definitions of ω_{*iT} and ω_{*i} .] Equations (9) – (11) are equivalent to the Sandberg 2-field model,² which is obtained by eliminating $\nabla_{\parallel} \delta J$ analytically. We do not perform that elimination here; instead, we retain the model in the 3-field form.

It turns out that the present toroidal ITG problem tests a generalized eigenvalue capability of the 2DX code. Specifically, of the three unknowns, $\delta\Phi$, δT_i and δJ in Eqs. (9) – (11), only $\delta\Phi$ and δT_i appear on the left-hand-side. This results in a generalized eigenvalue problem where the matrix on the left-hand-side is non-invertible. The 2DX code (and the underlying SLEPc⁴ eigenvalue solver) have no difficulty with this

structure. This observation significantly broadens the class of models that can be treated by 2DX.

C. The toroidal ITG benchmark test

We choose base case parameters for a closed flux surface edge plasma case to define values for n_e , T_e , T_i , L_n . Dimensional benchmark parameters for case #1 are: $\mu = Z = 1$ (Hydrogen), $n_e = 4.77 \times 10^{12} \text{ cm}^{-3}$, $L_n = 1.09 \text{ cm}$, $T_i = T_e = 17.0 \text{ eV}$, $L_{Ti} = 0.260 \text{ cm}$, $\kappa_n = 0.00497 \text{ cm}^{-1}$ ($\sim 1/R$), $B = 1.57 \times 10^4 \text{ G}$, $\rho_s = 0.0268 \text{ cm}$ (i.e. evaluated where we will apply local theory), $c_s = 4.03 \times 10^6 \text{ cm/s}$ (i.e. evaluated where we will apply local theory). The corresponding dimensionless benchmark parameters are: $k_b \rho_s = 0$ to 1 (will be scanned), $\varepsilon_n = 2 L_n \kappa_n = 0.0108$, $\tau = 1$, $\eta_i = 4.20$. (For these parameters, the Sandberg-dimensionless results can be converted to CGS units by multiplying the dimensionless k_b by 1830 to get n and multiplying the Sandberg γ by 3.69×10^6 to get 1/s.)

For a second case, case #2, which has a broader instability band in k_b , we artificially increase κ_n and hence ε_n by a factor of 10, $\kappa_n = 10 \times 0.00497 \text{ cm}^{-1}$ ($\sim 1/R$), $\varepsilon_n = 0.108$.

Figure 5 shows the comparison of 2DX code results with semi-analytic growth rates for benchmark cases #1 and #2.

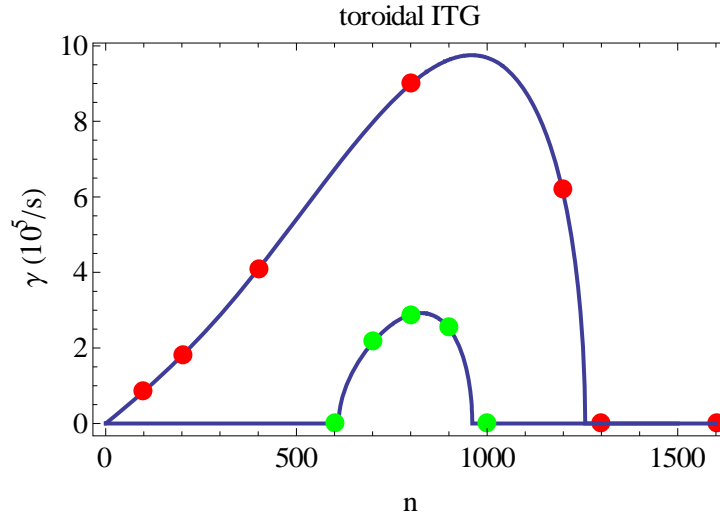


Fig. 1 ITG growth rate for the 2DX benchmark cases #1 (green) and #2 (red). Solid lines are the analytical result, disks are the 2DX results.

n	γ (10^5 s$^{-1}$) case #1	γ (10^5 s$^{-1}$) case #2
100	0.000	0.852
200	0.000	1.792
300	0.000	2.870
400	0.000	4.089
500	0.000	5.401
600	0.000	6.729
700	1.111	7.967
800	2.045	8.986
900	1.201	9.629
1000	0.000	9.685
1100	0.000	8.804
1200	0.000	6.073
1300	0.000	0.000

Table 1. Table of semi-analytic growth rates for the toroidal ITG benchmark cases shown in Fig. 1. These are the target results for the benchmark test of the numerical code (2DX).

References

1. see e.g. J. Chen and A.K. Sen, Phys. Rev. Lett. **72**, 3997 (1994) for a brief summary of the slab and toroidal branches.
2. I. Sandberg, Phys. Plasmas **12**, 050701 (2005); and Refs. therein.
3. D. A. Baver, J. R. Myra and M.V. Umansky, Comp. Phys. Comm. **182**, 1610, (2011).
4. <http://www.grycap.upv.es/slepc/>

ELM benchmark of the 2DX code

D. A. Baver and J. R. Myra

Lodestar Research Corporation, Boulder Colorado 80301

Maxim Umansky

Lawrence Livermore National Laboratory

1 Introduction

This test was devised to verify the ability of the 2DX eigenvalue code to solve a simple fluid model relevant to the stability threshold of edge localized modes in tokamaks. Since the functionality of the 2DX code depends on both the source code itself and the input file defining the system of equations to solve (structure file), this test demonstrates both. Moreover, since the structure file used for this test represents a subset of a more general 6-field model, many of the terms in that test are also verified.

This test compares 2DX results to BOUT++ simulations. Previous comparisons with BOUT++ are referenced for purposes of interpreting the results.

2 Description

2.1 Code structure

The 2DX code is a highly flexible eigenvalue solver designed for problems relevant to edge physics in toroidal plasma devices. Its flexibility stems from the use of a specialized input file containing instructions on how to set up a particular set of equations. Because of this, the 2DX code permits model equations to be changed without altering its source code. The drawback to this approach is that any change to the structure file represents a potential source of error, necessitating re-verification. This problem is offset by the fact that the source code remains unchanged, thus testing one structure file builds confidence in the underlying code that interprets the structure file. Also, structure files can be translated into analytic form, thus allowing the user to verify that the file contains the equations intended.

The structure file contains two main parts: an elements section, which constructs the differential operators and other functions used in a particular set of equations, and a formula section, which assembles these into an actual set

of equations. This separation means that elements can be recycled in other structure files. By testing one structure file, one builds confidence in the elements used in that file. The main source of error when switching to a different structure file then is in the formula section, which can be manually verified by translating into analytic form.

Regardless of the content of the structure file, the 2DX code is fundamentally a finite-difference eigenvalue solver. As such, it is subject to the limitations of any code of its type.

2.2 Model equations

For this test we use the following model equations [1]:

$$\gamma \nabla_{\perp}^2 \delta\phi = \frac{2B}{n} C_r n \delta T_i - \frac{B^2}{n} \partial \nabla_{\perp}^2 \delta A + i \frac{B k_b}{n} \delta A \partial_r \frac{J_{\parallel}}{B} \quad (1)$$

$$\gamma \delta T_i = -i \frac{k_b}{B} \delta\phi \partial_r T_i \quad (2)$$

$$\gamma \left(\frac{n}{\delta_{er}^2} \right) \delta A = -n \mu \nabla_{\parallel} \delta\phi \quad (3)$$

where

$$C_r = \mathbf{b} \times \boldsymbol{\kappa} \cdot \nabla = -\kappa_g R B_p \partial_x + i(\kappa_n k_b - \kappa_g k_{\psi}) \quad (4)$$

$$\nabla_{\perp}^2 = -k_b^2 - jB(k_{\psi} - i\partial_x R B_p)(1/jB)(k_{\psi} - iR B_p \partial_x) \quad (5)$$

$$\partial_{\parallel} Q = B \nabla_{\parallel} (Q/B) \quad (6)$$

$$\nabla_{\parallel} = j \partial_y \quad (7)$$

In this notation, κ_g is geodesic curvature, κ_n is normal curvature, k_b is binormal wavenumber, k_{ψ} is radial wavenumber. $R B_p$ is poloidal flux density, as poloidal flux is used as a radial coordinate, j is the inverse Jacobian $1/JB$ which is used to define the poloidal coordinate, and Q is any quantity. The above equations are normalized to Bohm units, i.e. all distances are in units of ρ_s and all time scales are in units of ω_{ci}^{-1} .

2.3 Boundary conditions

This test case uses phase-shift periodic boundary conditions in the parallel direction, and zero-derivative boundary conditions in the radial direction. The phase shift in the parallel direction is given by:

$$\delta Q(y=0) = \delta Q(y=2\pi) e^{-i2\pi nq} \quad (8)$$

This ensures toroidal and poloidal periodicity in the field-line following coordinate.

2.4 Profile setup

The formulas in Eq. 1-3 are normalized to Bohm units. Values are converted by dividing input distances by ρ_s , and input magnetic fields are in Tesla. Output eigenvalues are multiplied by ω_{ci} .

The geometry used is a thick annulus. This annulus contains a self-consistent equilibrium magnetic field, current profile, and pressure profile. This profile is based on a specific data file [2], further details of which are provided later in this document in Figs. 2-4.

3 Numerical results

The code was tested by sweeping toroidal mode number from 5 to 100 and comparing with BOUT++ results. Growth rates were normalized by dividing by ω_A . This is calculated from the reference values $B=1.9412991$ T, $R=3.49717$ m, and $n=1.01 \times 10^{20} m^{-3}$, yielding a value $\omega_A = 8.51707 \times 10^5$. The results of this test are shown in figure 1. In addition, the raw data is shown in table 1.

These results show good agreement between the two codes for low toroidal mode numbers. Significant disagreement occurs at higher mode numbers. However, the amount of disagreement decreases rapidly with higher radial resolution in the BOUT++ simulations. Moreover, a similar discrepancy pattern was noted in a previous comparison of BOUT++ to ELITE [1].

3.1 Convergence study

In order to estimate the accuracy of the previous results, and in order to estimate error scaling with resolution, a convergence study was done. In this study, the $n = 50$ mode was calculated on grids with varying resolution in n_x and n_y . Each of these parameters was varied in multiples of two, with n_x ranging from 64 to 1024, and n_y ranging from 16 to 64. For each value of n_y , Richardson extrapolation was performed in n_x in order to estimate a correct value and to calculate a power law for error scaling. The corrected values were then used to perform Richardson extrapolation a second time in n_y .

The results of this study are shown in Figs. 5-6. The n_x scan for $n_y=64$ yielded a power law of $\epsilon \propto n_x^{-2.64323}$ with other n_y values giving similar power laws. The n_y scan of the extrapolated values yielded a power law of $\epsilon \propto n_y^{-1.94741}$. The final extrapolated value (normalized to ω_A) is .363751.

References

- [1] B. D. Dudson *et al*, Comp. Phys. Comm. **180** (2009) 1467.

- [2] P. Snyder, data file cbm18-dens8-y064-x260.grd.pdb.v2.hdf

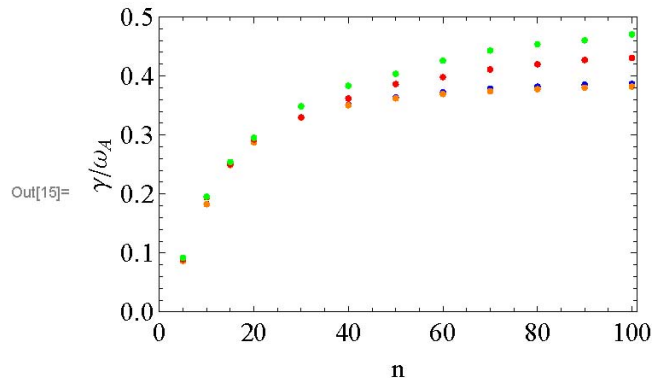


Figure 1: Growth rate vs. toroidal mode number for ELM models on 2DX and BOUT++. The blue dots are 2DX results for nx=512, the orange dots are 2DX results for nx=256, the red dots are BOUT++ results for nx=512, and the green dots are BOUT++ results for nx=256.

n	$Re(\gamma)/\omega_A$ 2DX (nx=512)	$Re(\gamma)/\omega_A$ 2DX (nx=256)	γ/ω_A BOUT++ (nx=512)	γ/ω_A BOUT++ (nx=256)
5	.0865383	.0864301	.0894333	.0916790
10	.182753	.182911	.194466	.195453
15	.248903	.249047	.250183	.254074
20	.287606	.287774	.292470	.295150
30	.329408	.329749	.330025	.348405
40	.350433	.351111	.361835	.383246
50	.362255	.363654	.386312	.403599
60	.369492	.372193	.398018	.426154
70	.374249	.378379	.410997	.443173
80	.377585	.381714	.419887	.453644
90	.380091	.385232	.427058	.460371
100	.382127	.386651	.430449	.470444

Table 1: Growth rate vs. toroidal mode number for 2DX vs. BOUT++

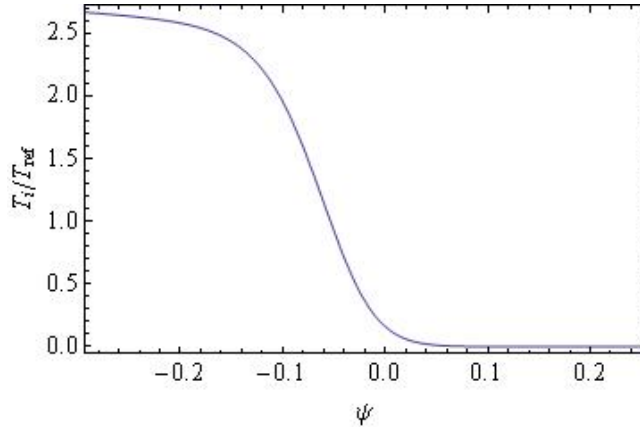


Figure 2: Ion temperature as a function of poloidal flux for the ELM profile.

Temperature is normalized to $T_{ref} = 635.2eV$. Note that these values are doubled compared to the original data file in order for the ion temperature equation to also capture the effects of electron pressure.

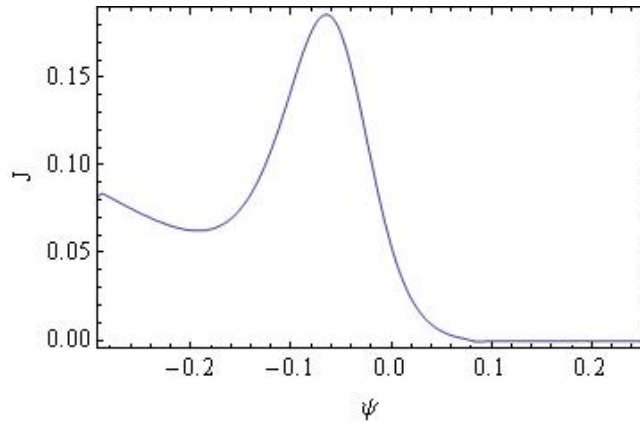


Figure 3: Current as a function of poloidal flux for the ELM profile. Current is normalized to $n_e c_s$ where $n_e = 1.01 \times 10^{14}$ and $c_s = 1.74143 \times 10^7$ in cgs units.

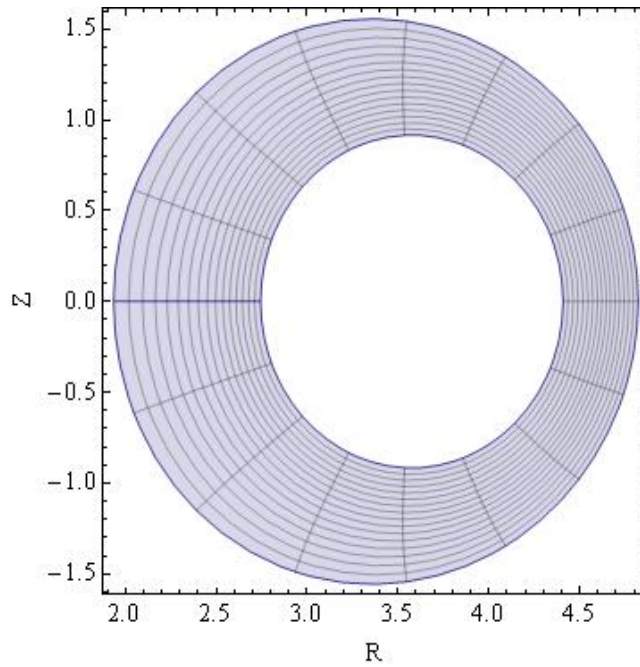


Figure 4: Flux surfaces in physical space for the ELM profile.

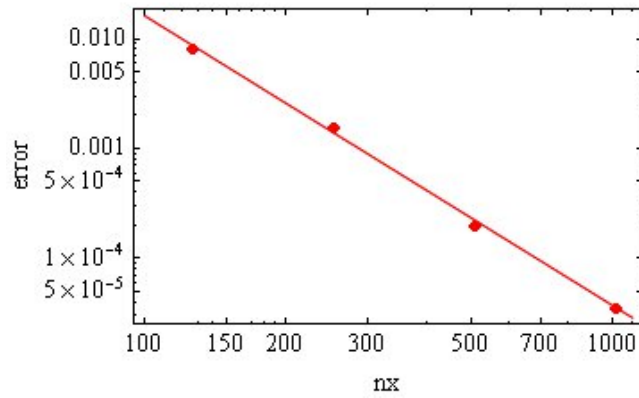


Figure 5: Convergence with increasing n_x for the ELM $n=50$ case at $n_y=64$

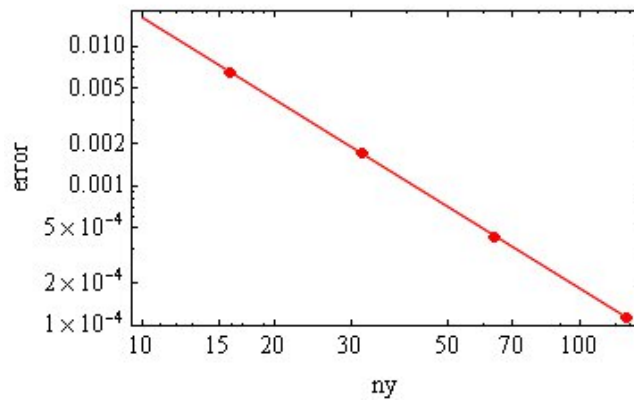


Figure 6: Convergence with increasing n_y for the ELM $n=50$ case.

Kinetic benchmark of the 2DX code

D. A. Baver and J. R. Myra

Lodestar Research Corporation, Boulder Colorado 80301

Maxim Umansky

Lawrence Livermore National Laboratory

1 Introduction

This test was devised to demonstrate and verify the ability of the 2DX eigenvalue code to solve a kinetic resistive ballooning model. Kinetic in this context refers to the inclusion of electron Landau damping into the basic physics model. Since the 2DX code is by nature a fluid code, this represents a significant extension of the code's capabilities. This extension is achieved through the use of an iterative method which progressively refines approximations to kinetic terms.

Fluid codes can be used to model kinetic effects using gyrofluid or other fluid moment models [1]. This offers a considerable advantage in computational cost compared to a fully kinetic model. However, such models are at best approximations, and in some cases involve non-analytic functions of wavenumber, hence cannot be expressed in sparse matrix form. The iterative method used in this report solves these problems by "tuning" the model equations to maximize accuracy for a specific eigenmode of interest.

This test compares 2DX results to results from a Mathematica test of a spectral kinetic model [2]. The spectral model uses an exact plasma response function, and therefore may be considered as a fully kinetic benchmark case.

2 Description

2.1 Code structure

The 2DX code is a highly flexible eigenvalue solver designed for problems relevant to edge physics in toroidal plasma devices. Its flexibility stems from the use of a specialized input file containing instructions on how to set up a particular set of equations. Because of this, the 2DX code permits model equations to be changed without altering its source code. The drawback to this approach is that any change to the structure file represents a potential source of error, necessitating re-verification. This problem is offset by the fact that the source

code remains unchanged, thus testing one structure file builds confidence in the underlying code that interprets the structure file. Also, structure files can be translated into analytic form, thus allowing the user to verify that the file contains the equations intended.

The structure file contains two main parts: an elements section, which constructs the differential operators and other functions used in a particular set of equations, and a formula section, which assembles these into an actual set of equations. This separation means that elements can be recycled in other structure files. By testing one structure file, one builds confidence in the elements used in that file. The main source of error when switching to a different structure file then is in the formula section, which can be manually verified by translating into analytic form.

Regardless of the content of the structure file, the 2DX code is fundamentally a finite-difference eigenvalue solver. As such, it is subject to the limitations of any code of its type.

2.2 Model equations

For this test we use the following model equations:

$$\gamma \nabla_{\perp}^2 \delta \phi = \frac{2B}{n} C_r \delta p - \frac{B^2}{b} \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (1)$$

$$\gamma \delta n = -\delta v_E \cdot \nabla n \quad (2)$$

$$-\gamma \nabla_{\perp}^2 \delta A = \nu_e \nabla_{\perp}^2 \delta A - \mu n \nabla_{\parallel} \delta \phi \quad (3)$$

$$\delta p = (T_e + T_i) \delta n + n(\delta T_e + \delta T_i) \quad (4)$$

$$C_r = \mathbf{b} \times \boldsymbol{\kappa} \cdot \nabla = -\kappa_g R B_p \partial_x + i(\kappa_n k_b - \kappa_g k_{\psi}) \quad (5)$$

$$\nabla_{\perp}^2 = -k_b^2 - jB(k_{\psi} - i\partial_x R B_p)(1/jB)(k_{\psi} - iR B_p \partial_x) \quad (6)$$

$$\partial_{\parallel} Q = B \nabla_{\parallel} (Q/B) \quad (7)$$

$$\nabla_{\parallel} = j \partial_y \quad (8)$$

$$\delta v_E \cdot \nabla Q = -i \frac{k_z (R B_p \partial_x Q)}{B} \delta \Phi \quad (9)$$

$$\nu_e = .51 \nu_r n / T_e^{3/2} \quad (10)$$

In this notation, κ_g is geodesic curvature, κ_n is normal curvature, k_b is binormal wavenumber, k_{ψ} is radial wavenumber. $R B_p$ is poloidal flux density, as poloidal flux is used as a radial coordinate, j is the inverse Jacobian $1/JB$ which is used to define the poloidal coordinate, and Q is any quantity. The above equations are normalized to Bohm units, i.e. all distances are in units of ρ_s and all time scales are in units of ω_{ci}^{-1} .

Kinetic effects are modeled by defining conductivity in terms of the plasma response function:

$$\nu_e = \mu/\sigma \quad (11)$$

$$\sigma = \frac{i\Omega_e}{\omega + i\nu_0} \zeta^2 Z'(\zeta) \quad (12)$$

$$\zeta = \frac{\omega + i\nu_0}{|k_{\parallel}| \sqrt{2} v_{the}} \quad (13)$$

where ν_0 is the physical resistivity, as opposed to ν_e which in these equations is used to model the effective resistivity.

Since the definition of ζ contains a non-analytic function of the wavenumber, it cannot be expressed in terms of spatial finite difference operators. In order to solve this problem, an iterative approach is used.

2.3 Iterative method

In order to calculate the plasma response function without using non-analytic functions of wavenumber, a polynomial approximation is used. In order to make this approximation as accurate as possible, the coefficients of this polynomial are iteratively updated to produce the most accurate possible fit at a particular wavenumber and growth rate. The growth rate for which the fit is to be optimized can be calculated using the eigenvalue of a previous iteration of the code, whereas the wavenumber can be calculated by doing simple numerical analysis of the eigenvector.

In this approach, conductivity is first represented as a polynomial in k :

$$\sigma = \frac{\mu}{\nu_e(a + bk^2)} \quad (14)$$

This formula can be represented in operator form by making minor modifications to Eq. 3. This yields:

$$-\gamma \nabla_{\perp}^2 \delta A = \nu_e a \nabla_{\perp}^2 \delta A - \nu_e b \nabla_{\perp}^2 \nabla_{\parallel}^2 \delta A - \mu n \nabla_{\parallel} \delta \phi \quad (15)$$

The coefficients a and b are then calculated using the plasma response function:

$$a = \frac{\beta k^3 \sigma_0}{2\alpha^2} \quad (16)$$

$$b = \frac{\sigma_0(2\alpha - \beta k)}{2\alpha^2} \quad (17)$$

$$\alpha = \frac{\Omega_e}{\nu + \gamma} \zeta^2 Z'(\zeta) \quad (18)$$

$$\beta = -\frac{\Omega_e}{\nu + \gamma} \zeta^3 Z''(\zeta) \quad (19)$$

The wavenumber k_{\parallel} can be extracted from the eigenvector by taking an average derivative:

$$k_{\parallel} = \sqrt{\frac{\sum_i j_i^2 |\psi_i - \psi_{i-1}|^2}{dy^2 \sum_i |\psi_i|^2}} \quad (20)$$

2.4 Boundary conditions

This test case uses phase-shift periodic boundary conditions in the parallel direction, and zero-derivative boundary conditions in the radial direction. The phase shift in the parallel direction is given by:

$$\delta Q(y = 0) = \delta Q(y = 2\pi) e^{-i2\pi nq} \quad (21)$$

This ensures toroidal and poloidal periodicity in the field-line following coordinate.

2.5 Profile setup

The formulas in Eq. 1-3 are normalized to Bohm units. Values are converted by dividing input distances by ρ_s , and input magnetic fields are in Tesla. Output eigenvalues are multiplied by ω_{ci} . Resistivity is given by the formula:

$$\nu_r = \frac{\mu}{.51\sigma_0} \quad (22)$$

where

$$\sigma_0 = 1.96 \frac{\omega_{ce}}{\nu_{ei}} \quad (23)$$

The geometry used is an idealized toroidal annulus with major radius R , minor radius a , and thickness δa . The density profile is exponential with scale length L_n , and temperature profiles are flat. Curvature is assumed, and is given by:

$$\kappa_n = \frac{\cos(y)}{R} \quad (24)$$

The function q may be sheared, but shear is set to zero for the test case given. The value of this constant q is given in Sec. 3.

Parallel derivatives are calculated using the Jacobian factor $j = 1/qR$. Toroidal mode number is calculated by $n = k_z a / q_0$.

3 Numerical results

The code and iterative method were tested by sweeping the variable T_e from 10 to 200. Ten iterations were used for each value for the kinetic method. The results from the kinetic iterative method were compared to results from the fluid model. In addition, a spectral calculation was used to determine the solution for the kinetic case using the full Z function rather than an approximation [2]. The other parameters used were:

$$a = .75 \text{ cm}$$

$$\delta a = .3 \text{ cm}$$

$$R = 207.5 \text{ cm}$$

$$L_n = 4 \text{ cm}$$

$$Z_{eff} = 1$$

$$B = 3 \text{ T}$$

$$n_e = 10^{13} \text{ cm}^{-3}$$

$$m_i/m_p = 2$$

$$\mu = 3674.32$$

$$T_i = 0$$

$$\ln\Lambda = 24 - \text{Log}(n_e/T_e)$$

The results of this test are shown in Fig. 1. In addition, a table of the raw eigenvalue data is shown in table 2. This test compares the 2DX kinetic iterative method (yellow diamonds), the 2DX fluid model (green triangles), the spectral full kinetic model (blue circles), and the spectral fluid model (red squares). As can be seen from this data, there are some slight discrepancies between 2DX and the spectral method even in the fluid case. This indicates the relative limitations of comparing a spectral to a spatial model; in particular, the spatial model is subject to numerical dispersion due to finite resolution. The discrepancy between the iterative and spectral kinetic methods is slightly larger, but it is still small enough for the method to be useful. This discrepancy can be explained because the eigenmode is not a pure sinusoidal function, so it does not have a single wavenumber. The iterative method is designed to get as good as possible a fit to a full plasma response function over a range of wavenumbers, but there are limits to how good a fit is possible.

References

- [1] G. W. Hammett, F. W. Perkins, Phys. Rev. Lett. **25**, 3019 (1990).
- [2] 2DX Phase I final report

nx	2	ny	64
dx	$.3/\rho_s$	dy	.097331
γ	$\gamma(s^{-1})/1.437 \times 10^8$	n	5
μ	3674.32	ν_r	$\nu_{ei}\mu/1.96\omega_{ce}$
q	3.3	k_z	$22\rho_s$
j	$\rho_s 668.25$	k_ψ	0
κ_n	$\rho_s/202.5 \cos(y)$	κ_g	0
B	3	RB_p	1

Table 1: Non-dimensional profile functions and parameters used in the resistive ballooning test case, as a function of the dimensional input $k_z(cm^{-1})$.

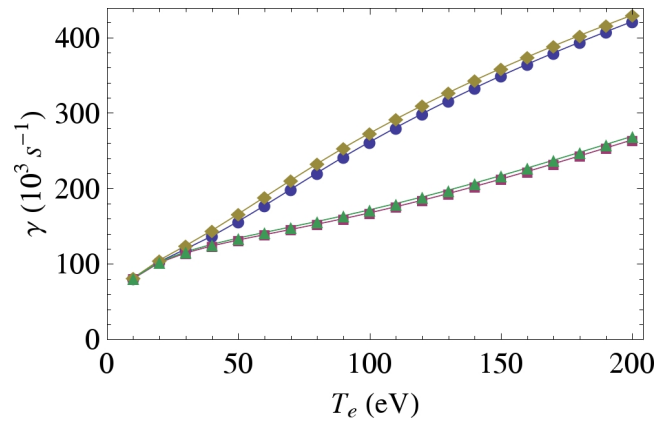


Figure 1: Growth rate vs. electron temperature for fluid and kinetic resistive ballooning models. Yellow diamonds are 2DX results using the iterative kinetic model. Green triangles are 2DX results using the fluid model. Blue circles are solutions to the kinetic model using a spectral method, and red squares are solutions to a fluid model using a spectral method.

T_e	γ (2DX kinetic)	γ (spectral kinetic)	γ (2DX fluid)	γ (spectral fluid)
10	81803	81079	81545	81007
20	105501	103325	103223	102137
30	124705	120139	116570	114943
40	144832	137440	126410	124286
50	166502	156785	134605	132036
60	188948	177778	142069	139109
70	211303	199440	149304	146005
80	232992	220954	156606	153013
90	253750	241840	164149	160306
100	273509	261883	172035	167982
110	292299	281027	180317	176090
120	310192	299295	189011	184644
130	327273	316746	198107	193630
140	343622	333449	207574	203015
150	359316	349472	217368	212751
160	374418	364882	227439	222781
170	388989	379736	237729	233047
180	403077	394085	248184	243489
190	416725	407975	258749	254053
200	429972	421445	269379	264688

Table 2: Growth rate vs. T_e for the kinetic and fluid resistive ballooning models.

Linear eigenvalue code for edge plasma in full tokamak x-point geometry

D. A. Baver and J. R. Myra,

Lodestar Research Corporation,

M.V. Umansky,

Lawrence Livermore National Laboratory

August 2010 (revised May 2011)

published in *Comp. Phys. Comm.* **182**, 1610, (2011)

DOE/ER/84718

LRC-10-137

LODESTAR RESEARCH CORPORATION

2400 Central Avenue
Boulder, Colorado 80301

Linear eigenvalue code for edge plasma in full tokamak x-point geometry

D. A. Baver and J. R. Myra

Lodestar Research Corporation, Boulder Colorado 80301

M. V. Umansky

Lawrence Livermore National Laboratory

A new code is presented for solving linear eigenvalue problems from fluid models of the edge plasma of tokamaks. The 2DX code solves linearized fluid equations in a 2D cross-section of the plasma, with toroidal mode number resolving the third dimension. Geometry capabilities include both closed and open field lines, allowing solution of x-point problems as well as a variety of other toroidal and cylindrical systems. The code generates a pair of sparse matrices forming a generalized eigenvalue problem which is then solved using a standard sparse eigensolver package. Use of a specialized equation parser permits a high degree of flexibility in both equations and coordinate systems. Both analytic and full geometry benchmark cases are presented.

1 INTRODUCTION

Fusion science in the 21st century is increasingly reliant on large scale computer simulations. This has led to an increasing need for verification and validation (V&V) [1]-[2] capability for large simulation codes.

Eigenvalue solvers for partial differential equations have a number of advantages as tools for verification of turbulence simulations. They are simpler than full nonlinear simulations, thus are inherently less prone to error and can be benchmarked adequately with a smaller and simpler set of test cases. They require significantly less computing resources, thus allowing tests to be done quickly and conveniently. Conversion of differential equations into matrix form can be separated from solution of eigenvalues, thus providing additional information for debugging purposes, as well as simplifying isolation of any errors that do arise.

The 2DX code is an eigenvalue solver designed to operate in an x-point topology. This makes it relevant to tokamak edge physics, particularly modeling the interaction between edge, scrape-off layer, and divertor plasmas. As such, it can be used to benchmark turbulence codes simulating that region of the plasma. In this paper, we present benchmarking of 2DX with the BOUT [4]-[5] edge turbulence code.

In addition to its application as a benchmarking tool, the versatility of the 2DX code makes it very useful for physics applications. The ability to calculate the growth rates of eigenmodes quickly makes it useful in determining stability thresholds for plasma instabilities such as ELM's [6]-[7]. Also, knowing the

spatial structure of dominant eigenmodes can provide insight into turbulence, for instance allowing estimation of typical frequency and wavenumber bands for fluctuations. In the direction parallel to the background magnetic field in the scrape-off-layer plasma, the degree of connection of modes between the x-point region and divertor plates is of interest [8]-[9]. An application of 2DX to this problem is discussed in Sec. 4.2.3.

An additional noteworthy feature of the 2DX code is its use of a specialized input format for parsing systems of equations. This gives the code an exceptional degree of flexibility in handling different physics models. It also offers a number of advantages from the standpoint of code verification. First, it splits each problem into two parts: the equation language file, and the source code to parse that equation language. While this introduces a potential source of error each time a new set of equations is used, this problem can be easily isolated, thus accumulating confidence in the source code across many different benchmark cases. Second, the equation language file can be translated into analytic form, thus allowing the user to determine, in an intuitive manner, precisely what equations are being solved. This offers a considerable advantage over codes in which equation sets are hard-wired into the source code, which is difficult to read, mingles formula and numerical technique, and lacks concise expression. Moreover, it maximizes transparency in the area where errors are most likely to occur, and does so in a way that is accessible to the casual user.

2 PROGRAM STRUCTURE

The 2DX code consists of two main parts, as well as a number of tools required to set up its input files and process its output files. The relation between these parts is shown in Fig. 1. The tools shown in this figure (grid generation tools, structure file viewer, data analysis tools) are currently implemented as Mathematica worksheets, although Python scripts for grid generation are also available. The 2DX code handles input and output as text files. The input file contains lists of constants and input functions (including Jacobian factors) on the grid. For simple problems and geometries, the input file can be created rather easily by a small separate code or script. For complex problems, such as the divertor geometry problem to be considered in Sec. 4.2.1, we employ a Mathematica notebook that calculates functions such as coordinate systems curvatures, shear, and Jacobians in toroidal geometry. This notebook can perform high order interpolations to refine a magnetic geometry mesh that originates from experimental equilibrium reconstructions.

The 2DX output file contains the specified number of eigenvalues and corresponding eigenfunctions on the mesh. These are extracted and processed by separate data analysis tools. Again, for simple problems and geometries, a small plotting code is all that is required. For complex problems in toroidal geometry, Mathematica or IDL codes provide more sophisticated tools that make use of the magnetic topology. Examples will be discussed later, in connection with Figs. 9,10 and 16. Finally, the Structure file viewer, and the associated procedure for creating structure files, will be discussed in Sec 2.4.

The principal components of the 2DX code are the core 2DX program and the eigenvalue solver. The core 2DX code generates a pair of sparse matrices in coordinate list format (sometimes referred to as COO format), i.e. as a tuple containing the (value, row, column) of each nonzero entry. These matrices form a generalized eigenvalue problem,

$$Ax = \lambda Bx \tag{1}$$

The pair of matrices is passed to an eigenvalue solver, which returns the eigenvalues λ and eigenvectors x . In the current version of the code, the eigenvalue solver employed is SLEPc [3]. This eigensystem package can attain a solution in a number of different ways, depending on user-selected options. Experience to date has achieved best results using a combination of a Krylov-Schur algorithm [10] combined with a Cayley spectral shift technique [11]. The spectral shift re-organizes the eigenvalues so that the eigenvalues with the largest real part (i.e. the fastest growing modes, hence the modes of interest) are also the largest absolute value eigenvalues. This is important because in a typical eigenvalue problem arising from the solution of partial differential equations, the reverse will be true: the largest absolute value eigenmodes will tend to be high wavenumber, poorly resolved modes that are either strongly damped or are neutrally stable with high frequency. Since sparse eigenvalue solution techniques lose their advantage if more than a handful of eigenvalues are returned, and since full eigenvalue solution techniques are impractical for large problems due to unfavorable scaling, it is critically important to choose a solution method

that prioritizes the modes of interest correctly.

Most of the distinguishing features of 2DX are in the core 2DX program, and relate to how it sets up the eigenvalue problem. Matrices are built up from built-in finite difference operators, boundary condition operators, and diagonal matrices constructed from functions. These simple matrices are combined into more complicated matrices using matrix addition and multiplication operations. The sequence of these operations is controlled by a specialized equation parser, using input in a data format called the equation language. The equation language is capable of immense versatility; given a sufficiently large set of equation language instructions, virtually any finite difference method of finite order or virtually any boundary condition can be created from these basic building blocks. This type of construction makes the code exceptionally flexible in what types of problems it can solve or what numerical methods it can use. In addition, it means that the 2DX source code itself contains only instructions for creating elementary operator matrices, performing elementary matrix operations, and parsing equation language files. This results in a code that is short, simple, and easy to debug. The drawback to this approach is that it shifts much of the burden of debugging to the various input files and the tools used to create these. This drawback is addressed subsequently. The net benefit of this approach is that 2DX possesses a modular structure that cleanly separates these tasks.

2.1 Elementary operators

Matrices in 2DX are built up from elementary operators. In addition to diagonal matrices used to represent profile functions or other functions derived from them, there are four finite difference operators and up to six boundary operators. The elementary differential operators (denoted u and l for upper and lower) are:

$$(\partial_x^u)_{ix,iy;jx,jy} = \begin{cases} -\frac{1}{dx} & \text{if } ix = jx, iy = jy \\ \frac{1}{dx} & \text{if } ix + 1 = jx, iy = jy \end{cases} \quad (2)$$

$$(\partial_x^l)_{ix,iy;jx,jy} = \begin{cases} \frac{1}{dx} & \text{if } ix = jx, iy = jy \\ -\frac{1}{dx} & \text{if } ix - 1 = jx, iy = jy \end{cases} \quad (3)$$

$$(\partial_y^u)_{ix,iy;jx,jy} = \begin{cases} -\frac{1}{dy} & \text{if } ix = jx, iy = jy \\ \frac{1}{dy} & \text{if } ix = jx, iy + 1 = jy \end{cases} \quad (4)$$

$$(\partial_y^l)_{ix,iy;jx,jy} = \begin{cases} \frac{1}{dy} & \text{if } ix = jx, iy = jy \\ -\frac{1}{dy} & \text{if } ix = jx, iy - 1 = jy \end{cases} \quad (5)$$

The boundary operators are matrices that are zero everywhere except at diagonal entries corresponding to grid points on a boundary. Thus, there are boundary operators for the upper and lower boundaries in x and y. In addition, there are two additional boundary operators that are offset by one grid cell from the upper boundaries in x and y. These are used to create boundary conditions on staggered grids.

From these basic operator matrices, it is possible to build up differential operators of arbitrary order through successive addition and multiplication. For

instance, suppose one wants to create a central difference second derivative operator. In this case, one would simply multiply two elementary operators: $\partial_x^2 = \partial_x^u \partial_x^l$. Likewise, a central difference first derivative can be constructed by adding two elementary operators and dividing by two: $\partial_x^c = (\partial_x^u + \partial_x^l)/2$.

In practice, most models allow for non-uniform grids. To account for this, the elementary operators must be multiplied by appropriate profile functions containing geometry information. This results in operators more complicated than the ones described above, but the overall concepts involved remain the same.

2.2 Staggered grids

An option in the equation language file is to make certain variables indented. This means that the last row of grid cells in one or both directions is deleted for that variable. The purpose of this is to permit staggered grids. Since grid points for that variable have one less row in one direction, they can be thought of as being between grid points on a normal grid. By choosing appropriate differential operators (upper or lower) in that direction for all terms linking the indented variable to non-indented variables and vice versa, one can make this concept a numerical reality.

The primary purpose of staggered grids in this code is to deal with numerical issues arising when a second derivative of the eigenfunction arises from two first derivative operators applied to different fields. For instance, as we will see in Eq. 27-32, the $\delta\phi$ and δA equations interact via parallel derivatives, so that

in the high collisionality limit there is an effective second parallel derivative on $\delta\phi$. If two central difference operators are used for the first derivative, the resulting second derivative operator skips directly adjacent grid cells. As a result, spurious eigenmodes with large k_{\parallel} emerge. Convolving first derivative operators that are offset in opposite directions yields the correct form for the second derivative operator, thus avoiding this problem. Given the underlying flexibility of the 2DX code, other applications of this capability (such as ensuring zero divergence of vector fields) are also possible.

2.3 Grid topology

In the present version of 2DX, the domain of the grid is divided into four regions. This feature can be easily generalized to handle more complicated topologies. The present form is suitable for a wide range of edge physics applications in tokamak divertor geometry.

One of these regions is the edge. This is the region inside the separatrix. It is bounded by periodic boundary conditions in y .

A second region is the scrape-off layer. This is the region outside the separatrix but adjacent to the edge. It is subject to sheath boundary conditions in y .

The other two regions are the private scrape-off layer. This is the region opposite the x-point from the edge. It is subject to both matching boundary conditions linking the two pieces of the private region, as well as sheath boundary conditions. In allocating space on the grid, it is located on either side of the

edge in y , and adjacent to the SOL in x .

The layout of these regions is shown in Fig. 2. Note that the x -point lies in the interstices between grid points, thus the singularity at the x -point is avoided. Because of the periodic boundary condition in the edge region, all eight grid points adjacent to the x -point connect to each other to form an octagonal cell, in contrast to the quadrilateral cells formed by other cycles of adjacent points.

2.4 Equation language

The equation language is a data file format containing all of the information the 2DX code needs to convert profile functions (i.e. temperature, density, magnetic geometry) into matrices to solve. This consists of a number of parts, of which three are of particular importance. These are the input language, the element language, and the formula language.

The input language defines the format of data files containing integer and real constants and profile functions. Here, the term profile function refers to an input quantity (for example a coefficient of the differential equations) that is specified on the 2D grid. Each entry in the input language consists of a data label and instructions on what to do with the actual data. This allows the 2DX code to determine whether a data block is supposed to contain an integer, a real number, or a profile function. Moreover, each data block is assigned a unique identifier so that it can be referenced by other parts of the equation language.

The element language consists of a series of basic operations that can be applied to profile functions or to operators. This is used to build up all of

the differential operators used in an equation set, as well as any derived profile functions (i.e. functions calculated from other functions). This is done by successive application of basic binary or unary algebraic or matrix operations. For algebraic operations, the current version of 2DX permits addition, subtraction, multiplication, division, powers, exponentials, and linear interpolation. For matrix operations, it permits addition, subtraction, multiplication, inverse, and transpose.

The formula language is used to specify the eigenvalue equations from the basic operators and functions created by the element language. Thus it generates the matrices required for a generalized eigenvalue problem. Since the equations of interest may contain more than one field, the eigenvector and matrices are larger than the number of grid points. That therefore means that the matrices will be larger than the elementary functions and differential operators created by the other parts of the language. To accommodate this, the 2DX code first multiplies together a string of functions and operators to form a matrix block. This block is then offset by adding integer multiples of its own size to the row and column indices of the elements of that block; this is straightforward to do for a sparse matrix in coordinate form, since row and column indices are simply integers associated with each nonzero entry. Adding to row indices determines which equation the term is in, whereas adding to the column index determines which field the term multiplies. By adding together a succession of such terms, the 2DX code is able to construct matrices to represent nearly arbitrary sets of equations provided they correspond to a linear eigenvalue problem.

The equation language can be converted via Python scripts between a numerical format that can be read by 2DX and a symbolic format that can be read by the user. In addition, a Mathematica script can be used to translate the numerical format into a standard algebraic form using symbolic logic.

For a simple example of how the equation language works, consider the following eigenvalue problem for $\delta\Phi$:

$$\frac{\partial}{\partial t} \nabla_{\perp}^2 \delta\Phi = \mu_{ii} \nabla_{\perp}^4 \delta\Phi \quad (6)$$

This can be coded using the symbolic format of the equation language as:

$$gg * dprp2 * PHI = mu_{ii} * dprp4 * PHI \quad (7)$$

where the element language contains instructions for building the operators dprp2 and dprp4 from elementary operators. This can then be translated into the numerical format and read by the Mathematica script to yield the following:

$$\nabla_{\perp}^2 \lambda \delta\Phi = \mu_{ii} \nabla_{\perp}^4 \delta\Phi \quad (8)$$

where λ is the eigenvalue. In this particular case, the element language is only used to define operators. In cases where the element language is used to define functions, the equation viewer will unravel these equations so as to display algebraic expressions of profile functions whenever possible.

The combination of symbolic format structure files and the Mathematica viewer script provides a layer of protection against coding errors in the equation language. Since the viewer script uses the same parsing logic as the 2DX code, it therefore displays the equations as they will actually be run. Thus, if a term

in the structure file is coded incorrectly, a discrepancy between the input and viewer output will become evident.

3 PHYSICAL MODEL

3.1 Coordinate geometry and differential operators for tokamak edge plasma

The equation language files used in the test cases presented later in this article are based on a ballooning (field line following) coordinate system. This coordinate system is also used by BOUT[4]-[5]. This coordinate system is defined by:

$$x = \psi - \psi_s \tag{9}$$

$$y = \theta \tag{10}$$

$$z = \zeta - \int_{\theta_0} d\theta \nu(\psi, \theta) \tag{11}$$

where ζ is the toroidal angle, θ is the poloidal angle, ψ is poloidal flux, and ν is the local safety factor.

Using toroidal symmetry of the equilibrium, we assume that the solution is periodic in ζ . Then, by specifying the eigenfunction in the form:

$$\delta\phi = \phi_1(x, y)e^{inz} \tag{12}$$

the potentially rapid (for large n) phase variation is extracted and the numerics need only resolve the coefficient ϕ_1 . Although closely related to the eikonal approximation [23] (a local approximation valid for $n \gg 1$, where ϕ_1 can be taken as a function of y alone), Eq. 12 and the numerical formulation can in principle be made exact (for infinite grid resolution).

The magnetic field can be calculated from the coordinates by introducing a local "safety factor" ν . This results in the following definition of B :

$$\mathbf{B} = \nu \nabla \psi \times \nabla \theta + \nabla \zeta \times \nabla \psi \quad (13)$$

From this field-line following coordinate system we derive formulas for parallel and perpendicular gradients based on coordinate derivatives. For this purpose we must construct profile functions containing relevant information about the structure of the magnetic field. The principal geometry profile functions used are:

$$j = \nabla \psi \times \nabla \theta \cdot \nabla \zeta / B \quad (14)$$

$$RB_p = |\nabla \psi| \quad (15)$$

$$k_b = -nB/RB_\theta \quad (16)$$

$$k_\psi = -nRB_p \left(\frac{\nu \nabla \theta \cdot \nabla \psi}{RB_p^2} + \int_{\theta_0} \frac{\partial}{\partial \psi} \nu \right) \quad (17)$$

$$\kappa_g = \kappa \cdot \hat{\mathbf{b}} \times \hat{\mathbf{e}}_\psi \quad (18)$$

$$\kappa_n = \kappa \cdot \hat{\mathbf{e}}_\psi \quad (19)$$

where

$$\kappa = \hat{b} \cdot \nabla \hat{b} \quad (20)$$

$$\hat{e}_\psi = \frac{\nabla \psi}{|\nabla \psi|} \quad (21)$$

and B_θ is the poloidal field. The Jacobian quantities J and RB_p define the physical distances corresponding to the grid spacing, the quantities k_b and k_ψ define the eikonal wavenumbers orthogonal to the magnetic field in the $\hat{b} \times \hat{e}_\psi$ and \hat{e}_ψ directions [23], and the quantities κ_g and κ_n are components of field-line curvature.

Making the simplifying but non-essential assumption that parallel derivatives are small compared to perpendicular derivatives, we then derive the following basic operators:

$$\nabla_{\parallel} = J \frac{\partial}{\partial y} \quad (22)$$

$$\nabla_{\perp}^2 = -k_b^2 - BJ(k_\psi - i\partial_x RB_p) \frac{1}{BJ} (k_\psi - iRB_p \partial_x) \quad (23)$$

$$C_r = -\kappa_g RB_p \partial_x + i\kappa_n k_b - i\kappa_g k_\psi \quad (24)$$

Of these operators, the curvature operator C_r is constructed by averaging the operators ∂_x^u and ∂_x^l to give a central difference first derivative operator with second order accuracy. The operator ∇_{\perp}^2 alternates the use of ∂_x^u and ∂_x^l so as to yield a centered second derivative operator, also with second order accuracy. The operator ∇_{\parallel} is used in terms that link variables on the staggered grid to variables not on the staggered grid; for this reason, a single operator of the type ∂_y^u or ∂_y^l is sufficient to achieve a central difference first derivative

operator. Since operators of opposite types are used when going to and from the staggered grid, successive operations of this type alternate upper and lower derivatives so as to yield a centered second derivative operator.

3.2 Boundary conditions

Boundary conditions in the x direction are built into operators involving x derivatives, by adding suitable multiples of the boundary operators. For all of the test cases listed in the following section, the x boundary employs a zero derivative boundary condition. This is chosen in order to permit the 2D system to allow a 1D local limit, i.e. ϕ_1 constant in x, thus facilitating comparison to local analytic theory.

In the y direction, boundary conditions are more complicated. First of all, there are two types of boundary conditions: phase-shift periodic and sheath, with matching boundary conditions treated as a special case of phase-shift periodic. The layout of these is described in Fig. 2.

In the case of phase-shift periodic boundary conditions, the goal is to modify the differential operators so as to apply smoothly to the solution in field-line following coordinates, as described by the ϕ_1 term in Eq. 12. Since the periodic boundary condition also serves as a branch cut in z , it follows that a phase shift must be applied at the branch cut. Noting that:

$$\phi_1(y = 0) = \phi_1(y = 2\pi)e^{-2\pi inq} \quad (25)$$

and also noting that the normal off-diagonal terms of the ∂_y operators are equal to $1/dy$, it follows that the off-diagonal terms that need to be added

to the matrix to achieve the phase-shift periodic boundary conditions can be constructed by applying conjugation or sign inversion to terms of the form:

$$\frac{e^{2\pi i n q}}{dy} \quad (26)$$

Sheath boundary conditions depend on the underlying physics model. Because they can couple multiple fields, they are implemented as separate terms in the model equations rather than as modifications to existing operators. For instance, the boundary condition in a parallel current equation will typically depend on potential, temperature, and density. In these cases an appropriate formula is multiplied by a sum or difference of the built-in boundary condition operators. This approach permits the flexibility needed to simulate diverse physical boundary conditions.

3.3 6-field fluid model for collisional plasma

While the 2DX code is capable of solving eigenvalues for a wide variety of equation systems, actually exercising this capability represents a potential source of error. Each time a new equation set is converted into equation language form, there is a possibility that one of the instructions in that file is incorrect. To minimize this source of error, most of the test cases presented use subsets of a standardized physics model. The standardized physics model is a linearized version of BOUT equations [4]-[5], thus simplifying comparison between the two codes. The model equations are expressed in dimensionless Bohm units, where time is normalized to the ion cyclotron frequency Ω_i and length is normalized to the cold ion sound gyroradius $\rho_s = c_s/\Omega_i$ with $c_s^2 = T_e/m_i$. The full set of

model equations considered here is as follows:

$$\gamma \nabla_{\perp}^2 \delta \Phi = -i \omega_{*i} \nabla_{\perp}^2 \delta \Phi + \frac{2B}{n} C_r \delta p - \frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \delta A + \Gamma \nabla_{\perp}^2 \delta \Phi + \mu_{ii} \nabla_{\perp}^4 \delta \Phi \quad (27)$$

$$\gamma \delta n = -\delta v_E \cdot \nabla n + \frac{2}{B} (C_r \delta p_e - n C_r \delta \Phi) - n \partial_{\parallel} \delta u - \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (28)$$

$$\gamma \delta u = -\frac{1}{n} \nabla_{\parallel} \delta p - \frac{1}{n} \delta b \cdot \nabla p - \frac{2T_i}{B} C_r \delta u + \partial_{\parallel} \mu_{\parallel} \nabla_{\parallel} \delta u \quad (29)$$

$$\begin{aligned} \gamma \delta T_e &= -\delta v_E \cdot \nabla T_e - \frac{2(1.71)T_e}{3n} \partial_{\parallel} \nabla_{\perp}^2 \delta A - \frac{2}{3} T_e \partial_{\parallel} \delta u \\ &+ \frac{2}{3} \partial_{\parallel} \chi_{\parallel} (\nabla_{\parallel} \delta T_e + \delta b \cdot \nabla T_e) + \frac{4T_e}{3B} \left(\frac{1}{n} C_r \delta p_e - C_r \delta \Phi + \frac{5}{2} C_r \delta T_e \right) \end{aligned} \quad (30)$$

$$\begin{aligned} \gamma \delta T_i &= -\delta v_E \cdot \nabla T_i - \frac{2T_i}{3n} \partial_{\parallel} \nabla_{\perp}^2 \delta A - \frac{2}{3} T_i \partial_{\parallel} \delta u \\ &+ \frac{4T_i}{3B} \left(\frac{1}{n} C_r \delta p_e - C_r \delta \Phi - \frac{5}{2} C_r \delta T_i \right) \end{aligned} \quad (31)$$

$$\begin{aligned} \gamma \left(\frac{n}{\delta_{er}^2} - \nabla_{\perp}^2 \right) \delta A &= \nu_e \nabla_{\perp}^2 \delta A - \mu n \nabla_{\parallel} \delta \Phi + \mu T_e \nabla_{\parallel} \delta n + \mu T_e \delta b \cdot \nabla n \\ &+ 1.71 \mu n \nabla_{\parallel} \delta T_e + 1.71 n \mu \delta b \cdot \nabla T_e \end{aligned} \quad (32)$$

where γ is the eigenvalue with real part corresponding to growth rate, $\mu = m_i/m_e$, and δ_{er} is the reference electron skin depth. Other notations are standard (see e.g. Refs. [4],[5]). In addition, we define the following quantities:

$$\omega_{*i} = \frac{k_b \partial_r p_i}{nB} \quad (33)$$

$$\delta p = (T_e + T_i) \delta n + n(\delta T_e + \delta T_i) \quad (34)$$

$$C_r = \mathbf{b} \times \boldsymbol{\kappa} \cdot \nabla \quad (35)$$

$$\partial_{\parallel} Q = B \nabla_{\parallel} (Q/B) \quad (36)$$

$$\delta v_E \cdot \nabla Q = -i \frac{k_b (\partial_r Q)}{B} \delta \Phi \quad (37)$$

$$\delta \mathbf{b} \cdot \nabla Q = i \frac{k_b (\partial_r Q)}{\delta_{er}^2 B \mu} \delta A \quad (38)$$

The sheath boundary conditions for this model are as follows:

$$-\nabla_{\perp}^2 \delta A = -\sigma n (T_e + T_i)^{1/2} \left(\frac{1}{2} \frac{\delta T_e + \delta T_i}{T_e + T_i} - \frac{1}{2} \frac{\delta T_e}{T_e} + \frac{\delta \Phi}{T_e} \right) \quad (39)$$

$$\delta u = -\sigma \frac{\delta T_e + \delta T_i}{2(T_e + T_i)^{1/2}} \quad (40)$$

$$\delta T_e = \sigma \frac{4}{9} \frac{\chi_{\parallel}}{S_E T_e^{1/2}} (\nabla_{\parallel} \delta T_e + \delta \mathbf{b} \cdot \nabla T_e) \quad (41)$$

where $\sigma = \pm 1$.

Of the fields presented in this model, δu and δA are indented in y , i.e. they use a staggered grid in the y direction. Thus, whereas the operator ∂_{\parallel} used in Eqs. 27,28,30, and 31 uses the lower derivative ∂_y^l defined in Eq. 5, the operator ∇_{\parallel} used in Eqs. 29 and 32 uses the upper derivative ∂_y^u defined in Eq. 4. This is in addition to the indented fields being evaluated on one less row of grid cells in y than the other fields.

4 VERIFICATION AND TEST CASES

The 2DX code has been benchmarked against a number of different test cases. Initial tests (not shown here) successfully reproduced analytical solutions of the 2D quantum harmonic oscillator problem and served to verify the basic 2DX solver kernel. A suite of additional tests was developed to further test 2DX together with the structure file associated with the six-field model. These are described in detail in the present section.

Tokamak edge plasma combines complex magnetic geometry and very rich physics, even in the framework of a linearized fluid model. The full two-fluid

six-field model has numerous instability branches, which makes it challenging to verify the full model. A verification approach taken here is testing several selected instabilities standalone by reducing the full system to a particular subset of equation terms that support a selected branch. Most of these cases permit comparison to BOUT results. In addition, a number of them have analytic solutions that results from both 2DX and BOUT can be compared to.

The analytic test cases were chosen in order to provide a representative sample of instabilities commonly encountered in tokamak edge turbulence. The resistive ballooning mode is important in the edge and scrape-off layer, and the resistive drift wave is important in the plasma edge, particularly for steep profiles. The electromagnetic version of the drift wave also tests Alfvén wave physics, known to be important for edge plasmas. Ion temperature gradient modes are important throughout the plasma when the temperature profile varies more rapidly than the density profile. In addition, the geodesic acoustic mode was chosen in order to demonstrate the ability of the code and its equation set to model more complex observed physical phenomena that depend in an essential way on the toroidal geometry.

Thus the sequence of sub-models tests a large variety of physically relevant terms from the full six-field model. Importantly the tests also progress from simple slab geometries through more complex idealized tokamak models, and finally to full x-point divertor geometry. For full geometry tests, test models were chosen based on known instabilities relevant in their respective regions of interest. In addition, simple models were chosen in order to isolate any numerical

issues relating to the more complex topology or non-uniform grid these cases present.

4.1 Analytic tests

4.1.1 Resistive ballooning model

The resistive ballooning model [12]-[14] uses the following subset of the 6-field model:

$$\gamma \nabla_{\perp}^2 \delta \Phi = +\frac{2B}{n} C_r \delta p - \frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (42)$$

$$\gamma \delta n = -\delta v_E \cdot \nabla n \quad (43)$$

$$-\gamma \nabla_{\perp}^2 \delta A = \nu_e \nabla_{\perp}^2 \delta A - \mu n \nabla_{\parallel} \delta \Phi \quad (44)$$

For the analytic test case, the profile functions are set so as to create a homogenous problem; this simplifies calculation of analytic solutions. Specifically, the analytic geometry is a twisted annulus with a q of 1.5, and phase shift periodic boundary conditions in the parallel direction. Density has an exponential profile, and temperature is constant. Toroidal mode number is then calculated from the variable k_b by the formula $n = k_b a / q$ where a is the radius of the annulus.

The results from this test are compared to analytic approximations in the high and low k_z cases. In addition, this model has also been simulated using BOUT. This provides further validation of the 2DX code.

The results from this are shown in Fig. 3. Circles represent eigenvalues

calculated using 2DX. The curved lines are analytic approximations, and the crosses are growth rates calculated from BOUT with associated error bars. Since BOUT is a simulation and not an eigenvalue solver, the growth rates shown are calculated by fitting an exponential to fluctuation amplitude, and deviation from a pure exponential (for instance due to mode contamination) is used to calculate the error bars.

4.1.2 Resistive drift wave model

Two resistive drift wave models [15]-[16] were tested, one retaining only electrostatic terms, one including electromagnetic terms. The electrostatic model is as follows:

$$\gamma \nabla_{\perp}^2 \delta \Phi = -\frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (45)$$

$$\gamma \delta n = -\delta v_E \cdot \nabla n \quad (46)$$

$$-\gamma \nabla_{\perp}^2 \delta A = \nu_e \nabla_{\perp}^2 \delta A - \mu n \nabla_{\parallel} \delta \Phi + \mu T_e \nabla_{\parallel} \delta n \quad (47)$$

The electromagnetic model is as follows:

$$\gamma \nabla_{\perp}^2 \delta \Phi = -\frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (48)$$

$$\gamma \delta n = -\delta v_E \cdot \nabla n \quad (49)$$

$$\gamma \left(\frac{n}{\delta_{er}^2} - \nabla_{\perp}^2 \right) \delta A = \nu_e \nabla_{\perp}^2 \delta A - \mu n \nabla_{\parallel} \delta \Phi + \mu T_e \nabla_{\parallel} \delta n + \mu T_e \delta b \cdot \nabla n \quad (50)$$

Both of these models were tested in a slab geometry with periodic boundary conditions. Unlike the resistive ballooning model, the test case required some

special features to deal with properties of the model equations. In particular, these model equations yield the maximum growth rate when both k_x and k_y are large and in a particular proportion to one another; if growth rate is plotted as a function of these parameters, it displays a long ridge with height asymptotically approaching a maximum value. As a result, if high resolution is available in both the x and y directions, the model will produce a dominant eigenmode that is not well resolved in at least one of those directions. To solve this problem, resolution in the x direction is highly restricted, and the grid size is made very large so that the maximum resolvable derivative in that direction is small; the latter restriction is necessary because the 2DX code cannot handle resolution less than two in either direction. The resulting test case closely resembles a 1D model, and can therefore be easily compared to analytic theory. Moreover, it results in a dominant eigenmode that is well-resolved.

In this geometry, the model is compared to analytic theory and BOUT results for a number of values of k_{\parallel} , which is controlled by adjusting the size of the domain. In order to compare the model results without regards for dimensional quantities, the modes are de-dimensionalized by constructing variables as follows:

$$\omega_* = k_{\perp} v_{pe} \equiv k_{\perp} \frac{v_{te}^2}{\omega_{ce} L_n} \quad (51)$$

$$\sigma_{\parallel} = \left(\frac{k_{\parallel}}{k_{\perp}} \right)^2 \frac{\Omega_{ci} \omega_{ce}}{0.51 \nu_{ei}} \quad (52)$$

$$\sigma_{\perp} = 0.51 \nu_{ei} \mu \quad (53)$$

The results from this comparison are shown in Fig. 4 - 5.

For domains with low minimum k_{\parallel} , the dominant eigenmode will typically have a higher k_{\parallel} than the minimum allowed. This results in a spurious eigenmode that masks actual trends in growth rate as a function of k_{\parallel} . To solve this problem, the code is set to return a number of eigenvalues. Their associated eigenfunctions are then sorted by k_{\parallel} as measured by comparing phase angles of adjacent grid points. This permits the lowest k_{\parallel} eigenvalue to be extracted and compared to analytic theory.

4.1.3 Slab ion temperature gradient mode model

The slab ion temperature gradient mode model [17]-[18] uses the following subset of the 6-field model:

$$\gamma \nabla_{\perp}^2 \delta \Phi = -\frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (54)$$

$$\gamma \delta n = -n \partial_{\parallel} \delta u - \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (55)$$

$$\gamma \delta u = -\frac{1}{n} \nabla_{\parallel} \delta p \quad (56)$$

$$\gamma \delta T_i = -\delta v_E \cdot \nabla T_i - \frac{2}{3} T_i \partial_{\parallel} \delta u \quad (57)$$

$$-\gamma \nabla_{\perp}^2 \delta A = -\mu n \nabla_{\parallel} \delta \Phi + \mu T_e \nabla_{\parallel} \delta n \quad (58)$$

As with the resistive drift wave model, the ITG model is tested in a geometry with limited resolution in the x direction, in order to solve the equations in an effectively 1D limit. Unlike the resistive drift wave case, with the ITG test the code is set to return multiple eigenvalues, from which a parallel wavenumber

can be calculated from the spatial structure of the eigenmode. This allows an entire mode spectrum to be derived from a single run. More importantly, it returns some modes from the neutrally stable branch of the ITG solution.

The results from this test are compared to analytic theory as well as to runs from BOUT. The results from this are shown in Fig. 6.

4.1.4 Geodesic acoustic mode model

The geodesic acoustic mode model [19] uses the following subset of the 6-field model:

$$\gamma \nabla_{\perp}^2 \delta \Phi = +\frac{2B}{n} C_r \delta p - \frac{B^2}{n} \partial_{\parallel} \nabla_{\perp}^2 \delta A + \Gamma \nabla_{\perp}^2 \delta \Phi + \mu_{ii} \nabla_{\perp}^4 \delta \Phi \quad (59)$$

$$\gamma \delta n = +\frac{2}{B} (C_r \delta p_e - n C_r \delta \Phi) - n \partial_{\parallel} \delta u - \partial_{\parallel} \nabla_{\perp}^2 \delta A \quad (60)$$

$$\gamma \delta u = -\frac{1}{n} \nabla_{\parallel} \delta p \quad (61)$$

$$-\gamma \nabla_{\perp}^2 \delta A = \nu_e \nabla_{\perp}^2 \delta A - \mu n \nabla_{\parallel} \delta \Phi + \mu T_e \nabla_{\parallel} \delta n \quad (62)$$

The parameter Γ is set to a positive value in this model in order to provide an instability drive for the GAM. Physically, this represents coupling of the GAM to turbulence.

The geometry used in the GAM test is more complicated than in the previous test cases. This is because the GAM can only exist if there is geodesic curvature. Instead of a simple slab model, the GAM test is performed in an idealized torus. Also, instead of performing a scan in parallel wavenumber, a scan is performed instead in the q of the idealized torus, i.e. one with circular flux surfaces and

low inverse aspect ratio.

The results from this are shown in Fig. 7. This plot compares the results from the 2DX code to two different analytic solutions: one for the theoretical GAM neglecting mode coupling (see below), and one for a zonal flow. Since both modes are solutions of the model equation, and since the 2DX code returns the dominant eigenvalue, both solutions are needed for comparison. The zonal flow solution, which dominates at low q , matches the 2DX result in that regime, whereas the analytic GAM matches the 2DX result in the high q regime.

A peculiar anomaly in this test is the presence of regularly spaced deviations between the 2DX result and the analytic GAM. This is due to coupling with sound waves. Taking into account coupling between the GAM at frequency:

$$\omega_g^2 = \frac{1}{q^2 R^2} + \frac{2}{R^2} \quad (63)$$

and spatial harmonics of the sound wave at frequency:

$$\omega_s^2 = \frac{m^2}{q^2 R^2} \quad (64)$$

predicts mode coupling, and hence modification of the simple result, when:

$$q = \sqrt{\frac{m^2 - 1}{2}} \quad (65)$$

These results agree with the position of the deviations in Fig. 7.

4.2 Full geometry tests

4.2.1 DIII-D test cases

The 2DX code has been tested and cross-benchmarked against BOUT in a full geometry case based on the edge plasma of D-III-D [26]. The model tested was the resistive ballooning model. The pressure profile was modified somewhat in order to avoid having a dominant mode localized along the radial boundary, but the magnetic geometry is fully realistic.

To best verify the capabilities of the 2DX code, we sought a scalar parameter scan in which the eigenmodes vary from broad (to sample the magnetic geometry) to localized (to permit comparison with an local analytical theory). Such a test can be achieved by varying collisionality, through Z_{eff} . For relatively low values of collisionality, such as occur for these profiles with $Z_{eff} = 1$, and a toroidal mode number $n = 100$, the modes fill the torus (but are stronger on the outboard side). Increasing Z_{eff} to artificially large values for purposes of the verification test, we find eventual saturation of the growth rate with Z_{eff} as shown in Fig.8 as the modes collapse down to a local point in poloidal angle on the low field side. This full range of conditions is adequately benchmarked in this test between 2DX and BOUT. To obtain agreement for the low gamma, $Z_{eff}=1$ case, it was necessary to employ finite μ_{ii} , so that grid-induced dissipation in BOUT did not influence gamma (a rather low resolution was employed for these tests in BOUT).

Also shown in the figure are the 2DX results for $n = 10^3$ and 10^4 . As n

is increased the eigenfunctions also collapse in radial location, and the growth rate asymptotically converges to $4.83 \times 10^5/s$.

A simple local analytical result for the resistive ballooning mode is possible in the limit where Z_{eff} and n are both asymptotically large:

$$\gamma = c_s \sqrt{\frac{2}{R_{eff} L_n}} \quad (66)$$

This result, shown by a green dashed line in Fig. 8 is also at $4.83 \times 10^5/s$. Thus by a sequence of steps we have connected the BOUT and 2DX codes for realistic tokamak parameters to a solid analytical asymptotic result.

In addition to growth rates, eigenmode structures can also be compared. Figs. 9-10 show a comparison of eigenmode structures for this model for $Z_{eff}=1$. Fig. 9 is the 2DX result, while Fig. 10 is the BOUT result. While the two results are not entirely identical, this can be attributed to differences in grids, as well as differences in how the two codes handle branch cuts. In the latter case, such an issue can arise because 2DX applies a phase shift at the branch cut in Fourier space as described by Eq. 25, whereas BOUT applies this phase shift in real space as described in Ref. [4].

Finally, the $n=100$ $Z_{eff} = 1$ case, being representative of a realistic tokamak problem of research interest, was used to analyze the scaling of the computational cost and numerical accuracy of the 2DX code. A number of test cases were generated by interpolating the original profiles at varying resolution, and the run time and leading eigenmodes were compared. Relative error was calculated by assuming the correct value to be an asymptote to a power law fit. In

addition, the time required to set up the matrix was compared to the time spent by SLEPc to solve it. These results are shown in Fig. 11-12. We conclude that 2DX displays near-linear convergence with grid size for this problem, and that run time is dominated by the SLEPc eigenvalue solution. It should be noted that the SLEPc package is parallelized, but the present test was run in serial mode on a single Intel-PC processor.

4.2.2 LAPD test cases

A number of test cases were done in order to verify the potential of the 2DX code for physics applications relevant to the Large Plasma Device (LAPD) [25]. Since LAPD is an open field-line device with a straight magnetic field, generating the correct geometry for this is simply a matter of using the SOL region of the 2DX grid and applying geometric profile functions to create a cylindrical coordinate system. In this case the x direction corresponds to radius (or more accurately enclosed flux), and the y direction corresponds to the z direction in standard cylindrical coordinates. Azimuthal angle is handled by toroidal mode number.

On this grid, the electrostatic resistive drift wave model (see Sec. 4.1.2) was simulated using 2DX and compared to a 1D eigenvalue solver. Temperature profile was assumed to be flat, and density profile was assumed to be of the form:

$$n(r) = a_0 + a_1 \frac{(1 + a_4 x)e^x - e^{-x}}{e^x + e^{-x}} \quad (67)$$

$$x = \frac{a_2 - r}{a_3} \quad (68)$$

The results of this are shown in Figs. 13-14. From this it can be seen that the 2DX results and the 1D eigenvalue results are in reasonable agreement.

4.2.3 Connection length study

In the edge region of a tokamak plasma, resistive ballooning instabilities can give rise to coherent turbulent structures which are localized perpendicular to \mathbf{B} and extended along \mathbf{B} . These structures are often referred to as blobs or blob-filaments [20]. These structures ultimately carry energy to the divertor plates or walls of the device; hence, it is of interest to understand the parallel structure and length scale (i.e. connection length) of the filaments. Experiments have attacked this problem by looking for correlations between the fluctuations at the midplane, X-point and divertor regions [8]-[9]. Midplane to X-point correlations have been seen [8] while an examination of midplane to divertor correlations [9] suggests a connection only when the fluctuations are sufficiently far radially from the X-point, in agreement with some earlier analytical theory [21] and eikonal and numerical studies [23]-[24]. We show here that 2DX can shed insight on this problem. It is one example of a physics application of the 2DX code which utilizes the full divertor geometry capability.

We began with experimentally measured profiles and geometry for a discharge on the National Spherical Torus Experiment [22]. To model a blob filament, we introduced a local bump in the radial pressure profile. The radial location of this bump was varied as shown in Fig. 15 with case a) on the closed surfaces, case b) on the separatrix, and case c) entirely in the open field line

region. The resulting eigenmode structures for each case are shown in Fig. 16. These show an eigenmode that is stopped in the parallel direction at the X-point when the bump is on the separatrix, but is more extended in the parallel direction when the mode is radially further away from the x-point. Note in particular that in case b) the mode does not reach the divertor plate, but in case c) it does, as found in Ref. [9].

5 SUMMARY

A new eigenvalue solver, the 2DX code, has been developed. It is capable of solving 2D linear partial differential equations in an x-point, periodic, or open field line topology. While designed specifically for problems in plasma physics pertaining to the edge of tokamaks, it is an immensely flexible code capable of solving a wide variety of problems.

The 2DX code has been tested against a number of cases, both in simple analytic geometry and in magnetic geometry derived from plasma experiments. These tests have been compared to analytic expressions and simulations of BOUT. Both comparisons have produced positive results.

The 2DX code shows great potential both as a benchmarking tool for plasma turbulence simulations and for direct physics applications. In the former case, it provides a simple code to which more complex codes can be compared. In the latter case, the ability to determine the spatial structure and growth rates of dominant eigenmodes of a system without the computational cost of a full

simulation (typically tens of CPU-hours for BOUT compared to a few CPU minutes for 2DX) can provide useful insight even in turbulent systems where dominant eigenvalues alone do not fully characterize its behavior.

Acknowledgements

This work was supported by the U.S. Department of Energy under grant DE-FG02-07ER84718 and by LLNL under DOE contract DE-AC52-07NA27344.

References

- [1] X.Q. Xu and R.H. Cohen, *Contrib. Plasma Phys.* **36**, 158 (1998).
- [2] M.V. Umansky et al., *Contrib. Plasma Phys.* **44**, 182 (2004).
- [3] www.grycap.upv.es/slepc/
- [4] M.V. Umansky, X.Q. Xu, B. Dudson, L.L. LoDestro, J.R. Myra, *Computer Phys. Comm.* **180**, 887 (2009)
- [5] X.Q. Xu, et al., *Phys. Plasmas* **7** 1951, (2000).
- [6] P. B. Snyder, et al., *Nucl. Fusion* **47**, 961 (2007).
- [7] B. J. Burke, S. E. Kruger, C. C. Hegna, P. Zhu, P. B. Snyder, C. R. Sovinec and E. C. Howell, *Phys. Plasmas* **17**, 032103 (2010).
- [8] J. L. Terry, S. J. Zweben, M. V. Umansky, et al., *J. Nucl. Mater.* **390**, 339 (2009).
- [9] R. J. Maqueda, R. Maingi and NSTX team, *Phys. Plasmas* **16**, 056117 (2009).
- [10] V. Hernandez, J. E. Roman, A. Tomas, V. Vidal, SLEPc technical report 7.
- [11] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst (eds.) (2000). *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

- [12] H. Strauss, Phys. Fluids **24**, 2004 (1981).
- [13] T. C. Hender, B. A. Carreras, W. A. Cooper, J. A. Holmes, P. H. Diamond,
and P. L. Similon, Phys. Fluids **27**, 1439 (1984).
- [14] P. N. Guzdar and J. F. Drake, Phys. Fluids B **5**, 3712 (1993).
- [15] W. Horton, Rev. Mod. Phys. **71**, 735 (1999).
- [16] M. Wakatani and A. Hasegawa, Phys. Fluids **27**, 611 (1984).
- [17] B. Coppi, M.N. Rosenbluth and R.Z. Sagdeev, Phys. Fluids **10**, 582 (1967).
- [18] S. Hamaguchi and W. Horton, Phys. Fluids B **2**, 1833 (1990).
- [19] N. Winsor, J.L. Johnson and J.M. Dawson, Phys. Fluids **11**, 2448 (1968).
- [20] S. I. Krasheninnikov, D. A. D'Ippolito, and J. R. Myra, J. Plasma Physics
74, 679 (2008).
- [21] R. H. Cohen and D. D. Ryutov, Contrib. Plasma Phys. **46**, 678 (2006).
- [22] M. Ono, et al., Nucl. Fusion **40**, 557 (2000).
- [23] J. R. Myra, D.A. D'Ippolito, X.Q. Xu and R.H. Cohen, Phys. Plasmas **7**,
2290 (2000).
- [24] X.Q. Xu, R.H. Cohen, W.M. Nevins, G.D. Porter, M.E. Rensink, T.D.
Rognlien, J. R. Myra, D.A. D'Ippolito, R.A. Moyer, P.B. Snyder and T.N.
Carlstrom, Nucl. Fusion **42**, 21 (2002).

- [25] W. Gekelman, H. Pfister, Z. Lucky, J. Bamber, D. Leneman, and J. Maggs,
Rev. Sci. Instrum. **62**, 2875 (1991).
- [26] J. L. Luxon, Nucl. Fusion **42**, 614 (2002).

Figure captions

Figure 1: Flowchart of data flow through the 2DX package. Modularization of the code into source code proper (2DX and SLEPc) as distinct from geometry (grid file) or equation set (structure file) has enabled rapid development and verification of the code.

Figure 2: Layout of the 2DX grid topology in the case of a single-null diverted tokamak. Cells that are adjacent on this chart are treated as adjacent by the code with the exception of cells adjoining the blue or purple lines; these are subject to a wraparound condition making them adjacent to more distant cells as indicated by the arrows.

Figure 3: Growth rate vs. k_b for resistive ballooning model. Black dots represent 2DX results, blue crosses represent BOUT results. The red curves are analytic approximations for large or small k_b , whereas the dashed line is an asymptotic solution for large k_b .

Figure 4 Growth rate vs. σ_{\parallel} for resistive drift wave. The blue and purple lines are analytic solutions for the electrostatic and electromagnetic models, respectively. The blue and green dots are the fastest growing eigenvalues from a 2DX run, whereas the orange and red dots (obscured by the blue and green dots for large σ_{\parallel}) are the eigenvalues corresponding to the longest wavelength eigenmodes from the same run.

Figure 5 Frequency vs. σ_{\parallel} for resistive drift wave. The blue and purple lines are analytic solutions for the electrostatic and electromagnetic models, respectively. The blue and green dots are the fastest growing eigenvalues from a 2DX run, whereas the orange and red dots (obscured by the blue and green dots for large σ_{\parallel}) are the eigenvalues corresponding to the longest wavelength eigenmodes from the same run.

Figure 6: Growth rate vs. k_{\parallel} for ion temperature gradient mode model. Blue crosses are eigenvalues from a 2DX run as functions of the k_{\parallel} value calculated from each corresponding eigenmode. Red crosses are analytic solutions calculated at the same k_{\parallel} values as 2DX eigenmodes. The green circles are BOUT results.

Figure 7: Growth rate vs. q for geodesic acoustic mode model. The blue curve represents results from the 2DX code. The tan curve is an analytic solution for the GAM mode, whereas the red curve is an analytic solution for a zonal flow mode.

Figure 8: Growth rate vs. Z_{eff} for resistive ballooning mode in D-IIID edge geometry. These results are for mode number $n=100$. The black lines at the right hand side represent 2DX solutions for $\mu_{ii} = 0$ and $n = 10^2, 10^3$, and 10^4 . The dashed green line is an analytic solution assuming both Z_{eff} and mode

number are large.

Figure 9: Eigenmode structure from 2DX for resistive ballooning mode in D-IIID edge geometry. Dots indicate the positions of grid points in the 2DX mesh. Colors indicate absolute value of relative amplitude, whereas uncolored regions indicate values near zero.

Figure 10: Eigenmode structure from BOUT for resistive ballooning mode in D-IIID edge geometry. Dots indicate the positions of grid points in the BOUT mesh. Colors indicate absolute value of relative amplitude, whereas uncolored regions indicate values near zero.

Figure 11: Log-log plot of the relative error in the eigenvalues for a divertor-geometry solution of the resistive ballooning mode. Various resolutions for these cases with $n_x \times n_y$ grids are shown. The solid red line is a least-squares power law fit. The dashed black line is a $1/n$ power law fit.

Figure 12: Log-log plot of the (single processor) CPU time for various grid resolutions where $n = \sqrt{n_x n_y}$. The solid blue line is a least-squares power law fit to all the data. Red points indicate the square-grid runs. Green points indicate matrix set-up time which is more than an order of magnitude smaller than total computational time.

Figure 13: Growth rates as a function of mode number for LAPD eigenvalue scan. Red circles indicate 2DX results, blue circles indicate results from a 1D eigenmode solver.

Figure 14: Frequencies as a function of mode number for the LAPD eigenvalue scan. Red circles indicate 2DX results, blue circles indicate results from a 1D eigenmode solver.

Figure 15: Pressure profiles used in the 2DX/NSTX connection length study. The curves labeled a, b, and c represent different perturbed pressure profiles used to control the location of the dominant eigenmode.

Figure 16: Eigenmodes of the resistive ballooning model from the 2DX/NSTX connection length study. Red indicates high eigenmode amplitude, blue indicates low eigenmode amplitude. The eigenmodes a, b, and c are calculated using the corresponding pressure profiles from Fig. 15.

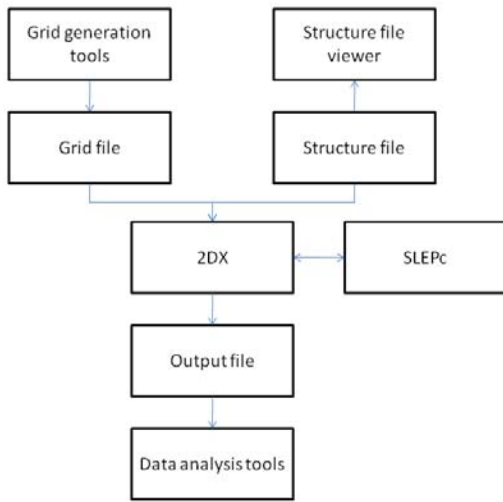


Figure 1:

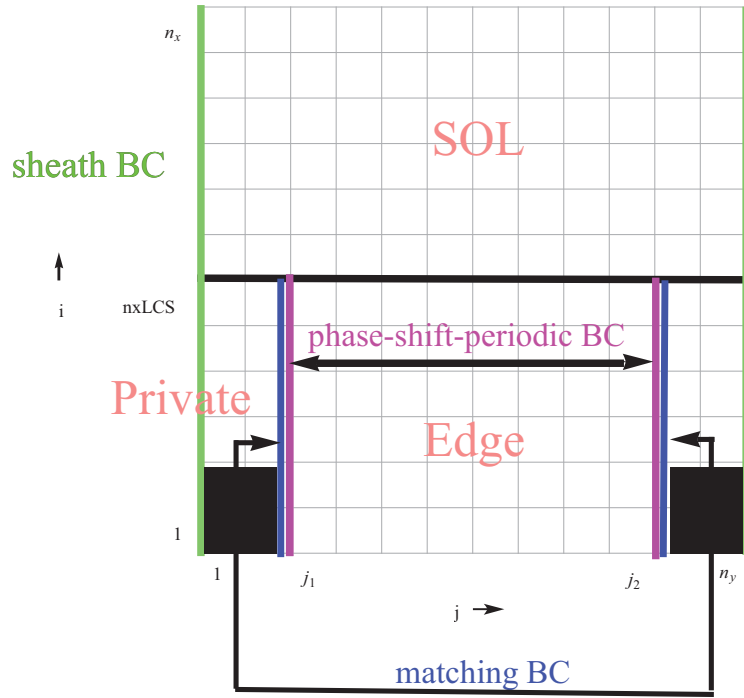


Figure 2:

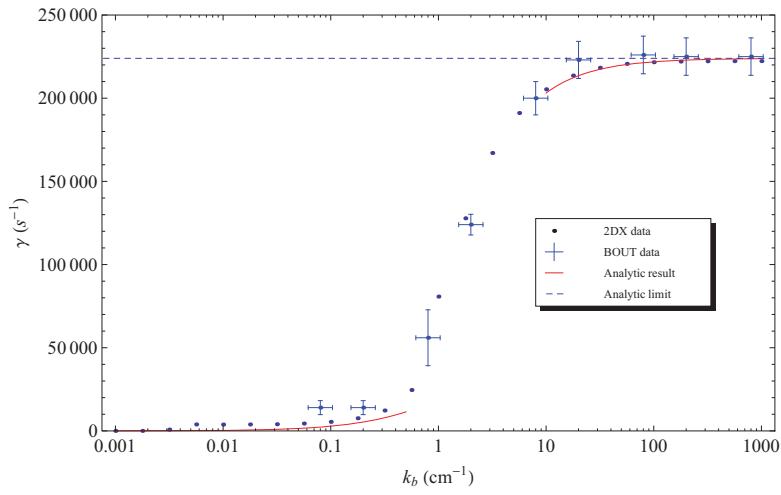


Figure 3:

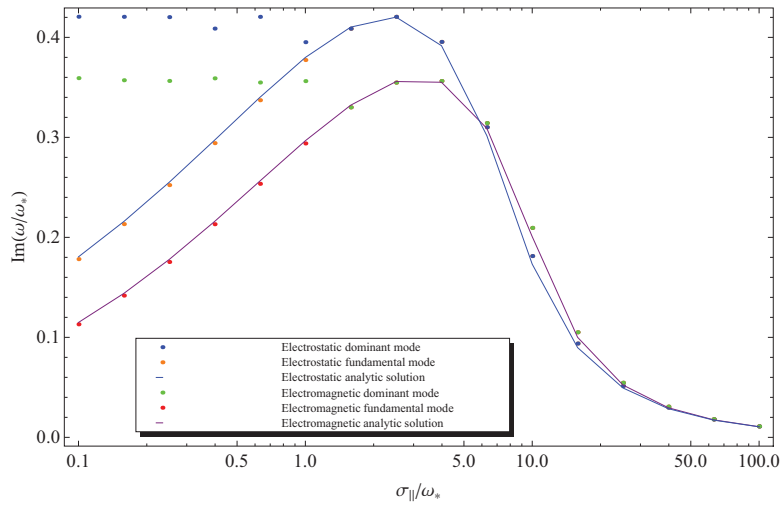


Figure 4:

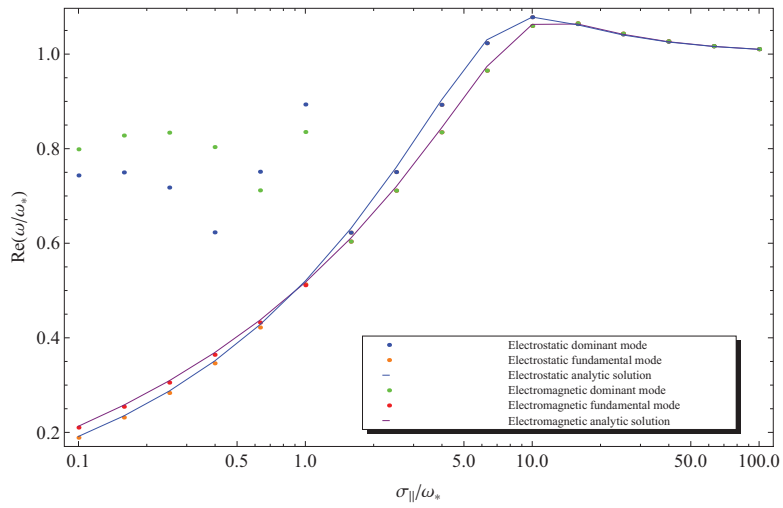


Figure 5:

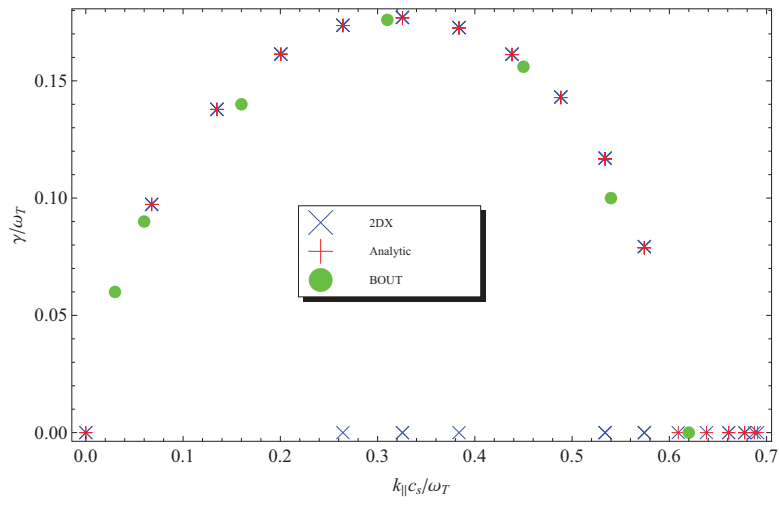


Figure 6:

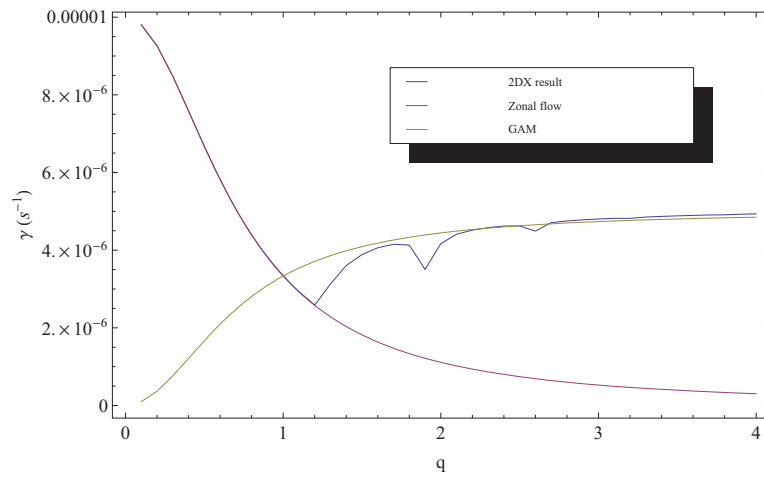


Figure 7:

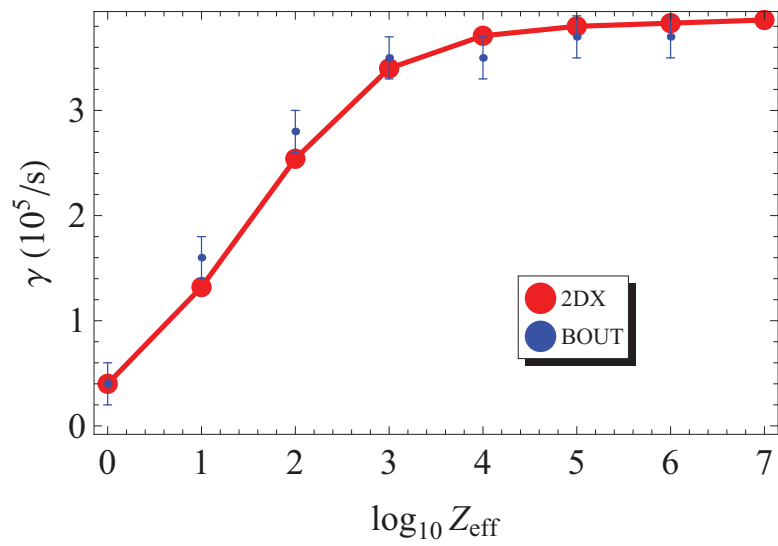


Figure 8:

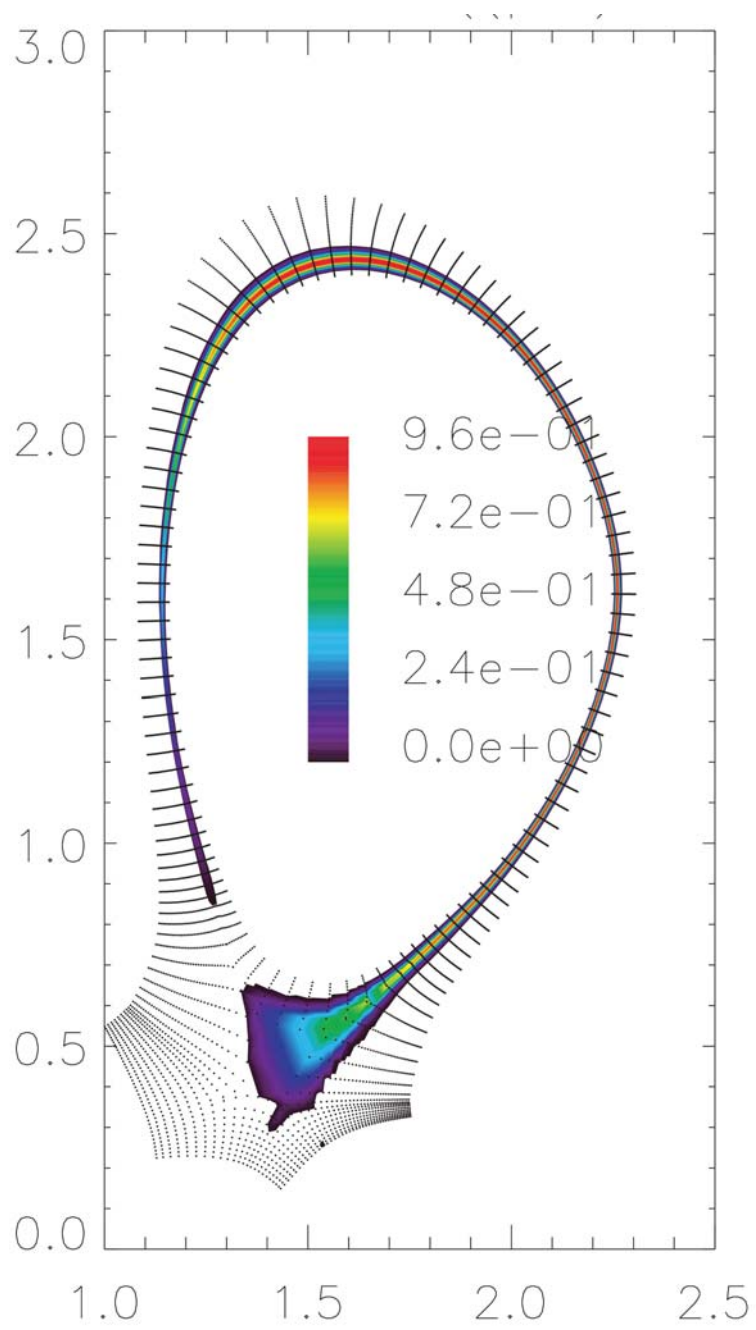


Figure 9:

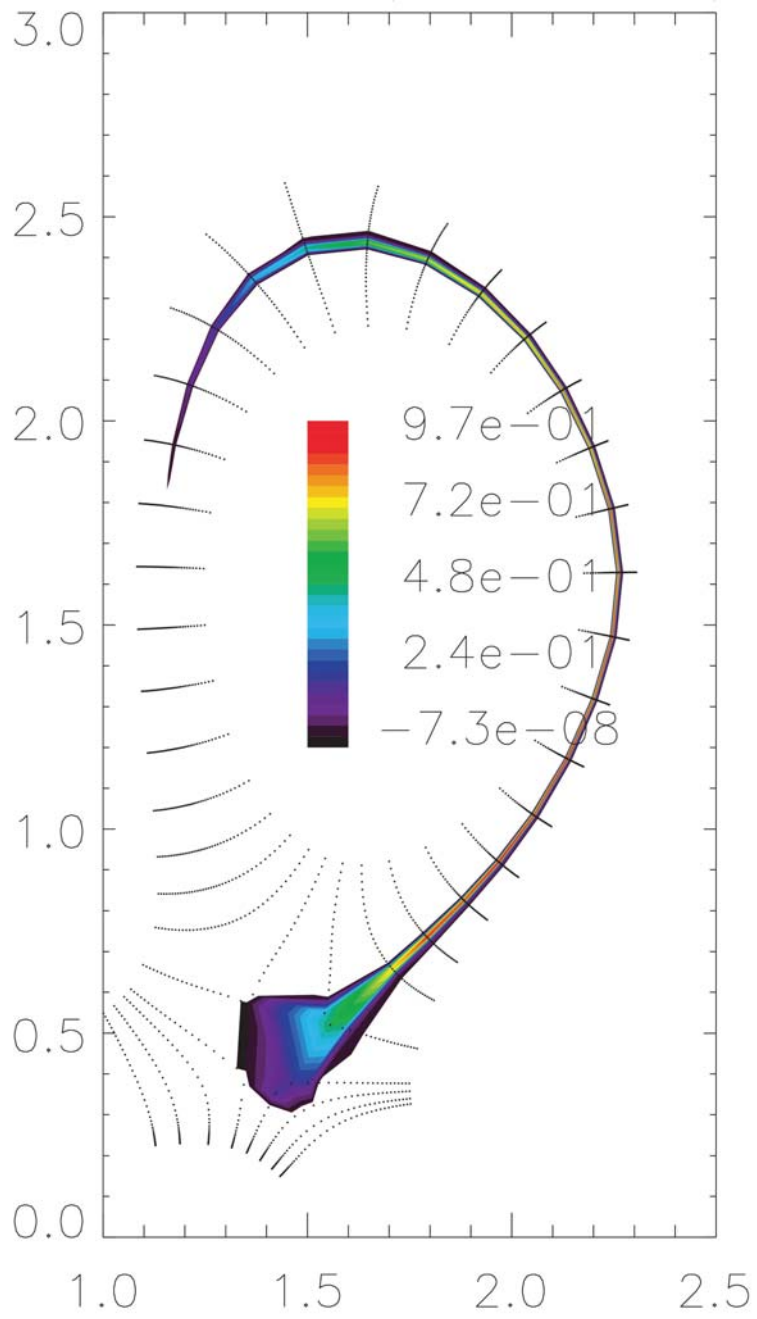


Figure 10:

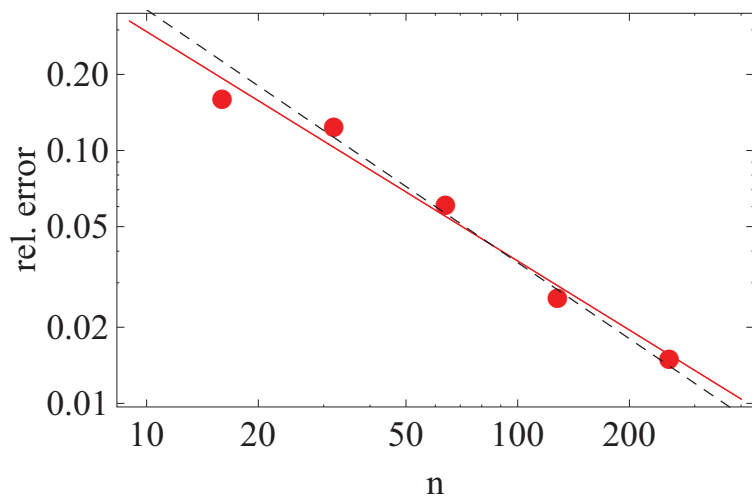


Figure 11:

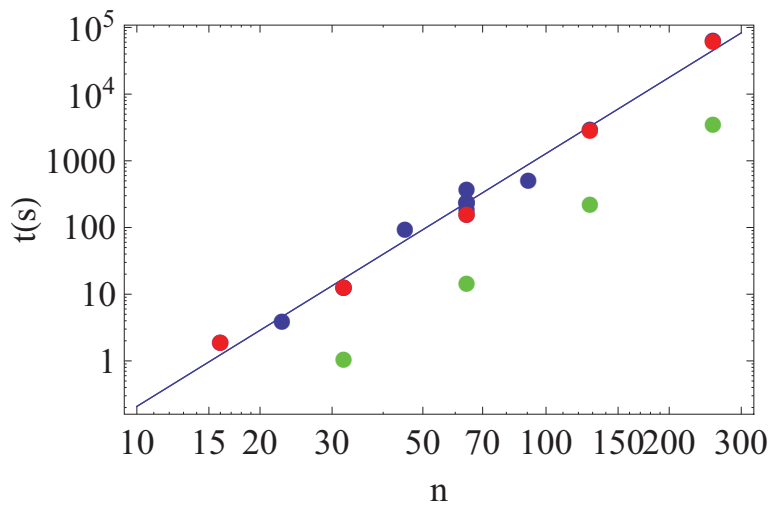


Figure 12:

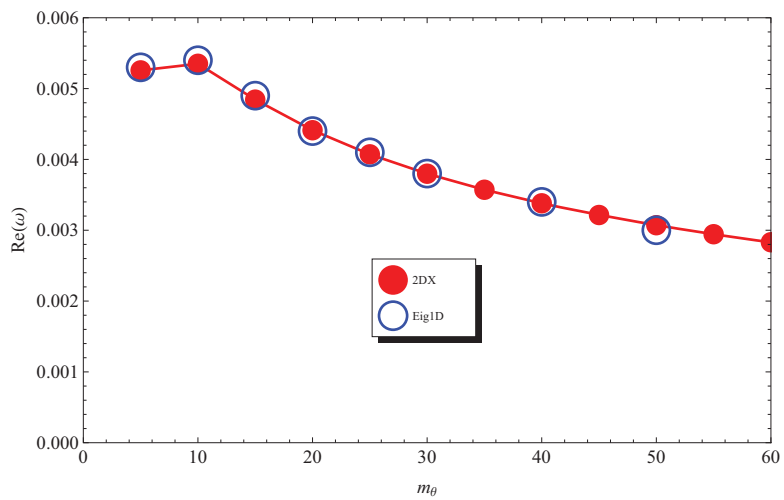


Figure 13:

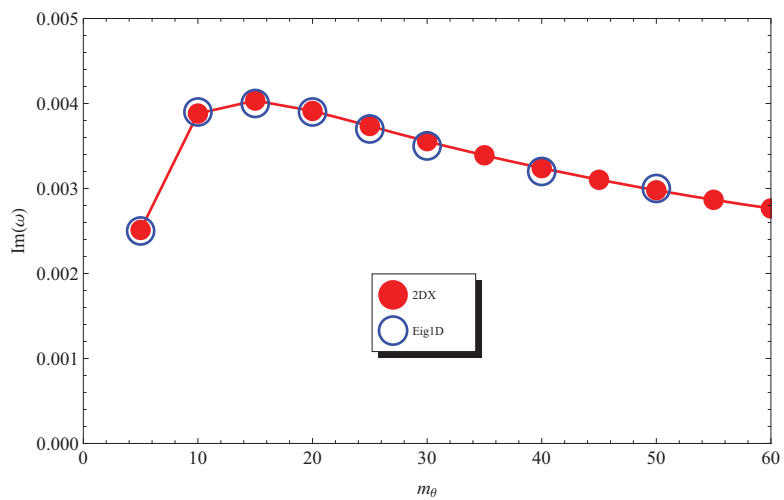


Figure 14:

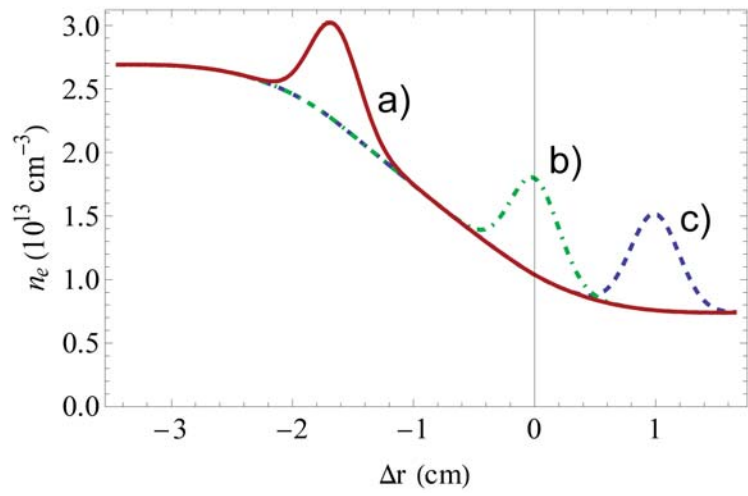


Figure 15:

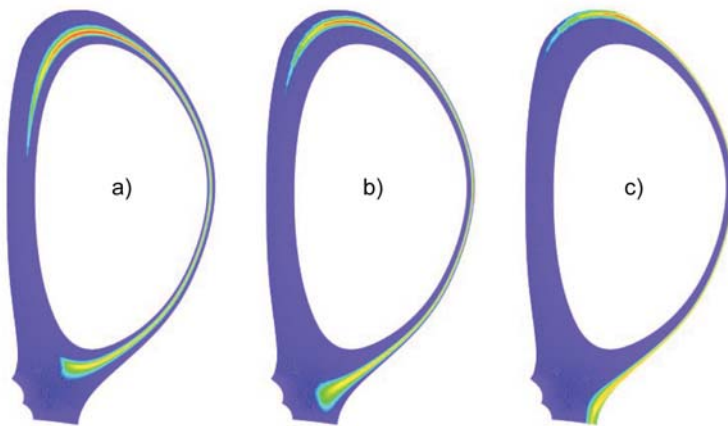


Figure 16: