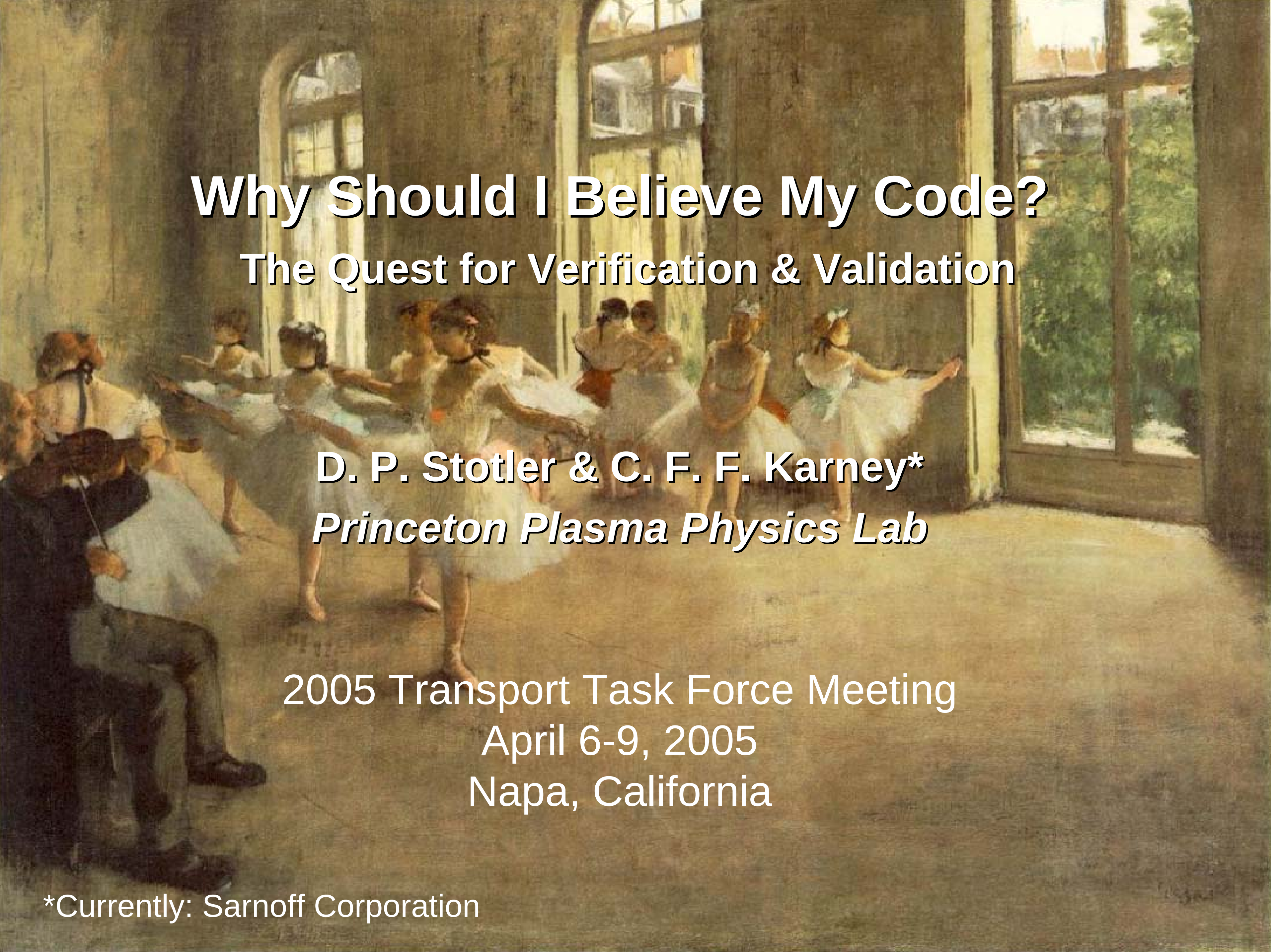


Why Should I Believe My Code?

The Quest for Verification & Validation

D. P. Stotler & C. F. F. Karney*

Princeton Plasma Physics Lab

2005 Transport Task Force Meeting
April 6-9, 2005
Napa, California

*Currently: Sarnoff Corporation

Outline

I. Introduction

II. DEGAS 2 Verification

   A. Code Verification

   B. Solution Verification

III. DEGAS 2 Validation

IV. Dedicated Validation Experiments

Introduction

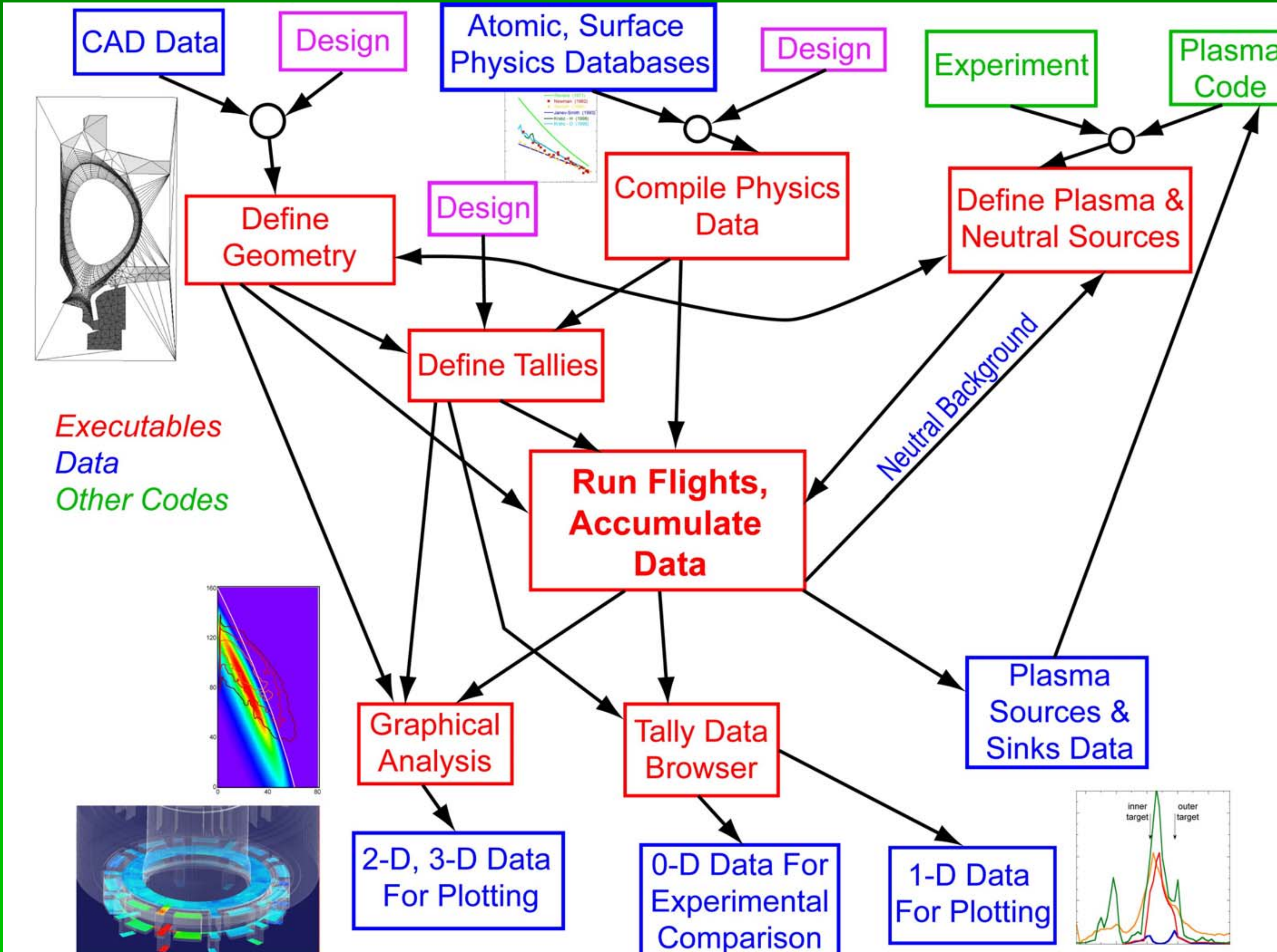
- **Scientists are supposed to be skeptical!**
  - **Treat the code as if it were a perpetual motion machine.**
  - **[Hatton 97]:** ***"…the results of scientific calculations carried out by many software packages should be treated with the same measure of disbelief researchers have traditionally attached to results of unconfirmed physical experiments."***
  - **Verification & validation methodologies facilitate this process.**
  - **These are ongoing efforts ⇒ a quest!**
- **Where are we going to get enough manpower & funding to do this?**
  - **Effort expended should be commensurate with risk associated with wrong solutions.**
  - **For most present fusion applications risks are not high,**
    - **Possible exception: ITER PFC decisions [Skinner, Edge & Pedestal II].**
  - **⇒ Might be able to get "good enough" with available resources.**
    - **But, should always keep in mind how we could do better.**
  - **Describe V&V of DEGAS 2 as an example,**
  - **Provide throughout ideas for further discussion.**

DEGAS 2 Uses Monte Carlo Algorithm to Solve Boltzmann Equation for Neutral Species [Stotler 94]

- More precisely, compute phase space integrals of form:

$$I(g) = \int_{\Delta \vec{x}} d^3x \int_{\Delta \vec{v}} d^3v \, f(\vec{x}, \vec{v}) g(\vec{x}, \vec{v})$$

- Two good features:
  - Treat complex geometry & processes,
  - Most aspects of code operation can be understood in analog terms,
    - But, have mathematical basis in central limit theorem, etc.
    - Needed also for non-analog techniques (variance reduction)
- Drawbacks:
  - Accuracy of solution scales like $N^{0.5}$,
    - Mitigated by natural parallelism.
  - Comparisons of solutions must be done statistically,
    - I.e., using a mean and a variance.


CAD Data
Design
Atomic, Surface Physics Databases
Design
Experiment
Plasma Code
Define Geometry
Design
Compile Physics Data
Define Plasma & Neutral Sources
Define Tallies
Neutral Background
Executables
Data
Other Codes
Run Flights, Accumulate Data
Graphical Analysis
Tally Data Browser
Plasma Sources & Sinks Data
2-D, 3-D Data For Plotting
0-D Data For Experimental Comparison
1-D Data For Plotting
inner target
outer target

Approach to V&V Dictated by Nature of Code Easier for MC?

- Modularity simplifies testing of components.
- MC code will always converge,
  - Only question is how fast.
- MC method naturally provides statistical error,
  - But, this is only 1 part of total simulation uncertainty!
- Flexibility facilitates simulating variety of verification problems & experimental situations,
  - In as much detail as needed.
  - Simulating diagnostic signals straightforward.

Outline

I. Introduction

II. DEGAS 2 Verification

  A. Code Verification

  B. Solution Verification

III. DEGAS 2 Validation

IV. Dedicated Validation Experiments

Code Verification Is About Preventing & Eliminating Bugs – DEGAS 2's Exterminator

- Use CVS to track code versions & branches.
- Document all non-trivial code.
- `implicit none` – use it, love it.
- Use "assertions" liberally.
- Let the computer do the grungy coding.
- Use a variety of platforms & compilers.
- Check every code change in a debugger.

What Did You Break This Time? Regression Testing

- Regression test suite: set of runs performed after code changes to identify problems caused by those changes,
  - Require results to match previous ones “exactly”,
  - Or that deviations are the ones expected from code modifications.
- Ideally, would test all aspects of code functionality.
- DEGAS 2 has 8 examples,
  - Most derived from test cases discussed here.
  - Not comprehensive, but do try to add one with each new code capability.
- `matchout` code used to compare output files,
  - Checks every array, printing largest absolute and relative difference,
    - In most cases, this is 0 (to 15 digits).
  - Does statistical comparison, using relative standard deviations in files, for cases in which only statistical agreement is expected.
- DEGAS 2 examples not ready for automation,
  - Not quick or easy to run.

Code Verification
Ideas for Going Further

1. Static deep-flow analysis tools,
   - Imagine a compiler in a really bad mood.
   - Source of Hatton's "12 serious faults / 1000 executable FORTRAN lines" [Hatton 97].
   - Commercial: QA C, QA FORTRAN. Looks for:
     - Coding defects not found by compiler,
     - Non-portable code,
     - Code likely to be difficult to maintain,
     - & Standards violations.
     - Outputs software metrics.
   - Freeware: ftnchek.
2. Dynamic testing,
   - Includes regression testing.
   - Coverage analysis – which lines of code are executed?
     - Appear to be commercial & freeware packages available.
     - Including Lahey-Fujitsu F95!
     - Can design test cases to hit certain sections of code: glass box testing.
   - Black-box testing: have someone else run the code!

Outline

I. Introduction

II. DEGAS 2 Verification

  A. Code Verification

  B. Solution Verification

III. DEGAS 2 Validation

IV. Dedicated Validation Experiments

Checking Code Components Separately Speeds & Simplifies Testing

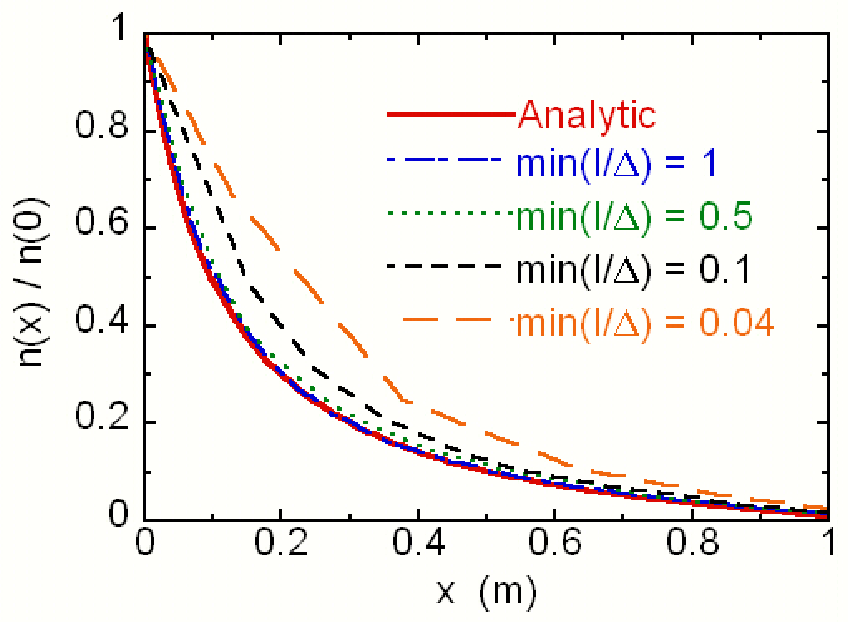
- Test code modifications or new atomic physics / PMI data in isolation,
- Check input data for new problems at each stage of workflow,
- DEGAS 2 has test routines for:
  - System-dependent routines,
  - Random number generator,
  - Atomic physics reactions,
  - Plasma-material interactions,
  - Geometry,
  - Sources.

Integrated Tests of DEGAS 2

- Against analytic solutions,
  - Escape probability,
  - 1-D analytic fluid model,
  - Couette flow.
- Against other codes, aka "benchmarks",
  - DEGAS, EIRENE, KN1D
  - Checks numerical methods used in both codes,
  - And, gives some idea of impact of different approaches.
- Some of these are suitable for inclusion in a public collection of test cases,
  - Certainly for Monte Carlo neutral transport codes,
  - Some are in fluid limit $\Rightarrow$ test fluid neutral transport codes.
- Conservation checks are also legitimate verification tests,
  - Not yet automated in DEGAS 2, but moving towards that.
  - Also symmetry tests.
- Many simpler checks carried out, but not well documented,
  - Whenever I see a viable test, I try it!
  - Going forward: compile a record of these.

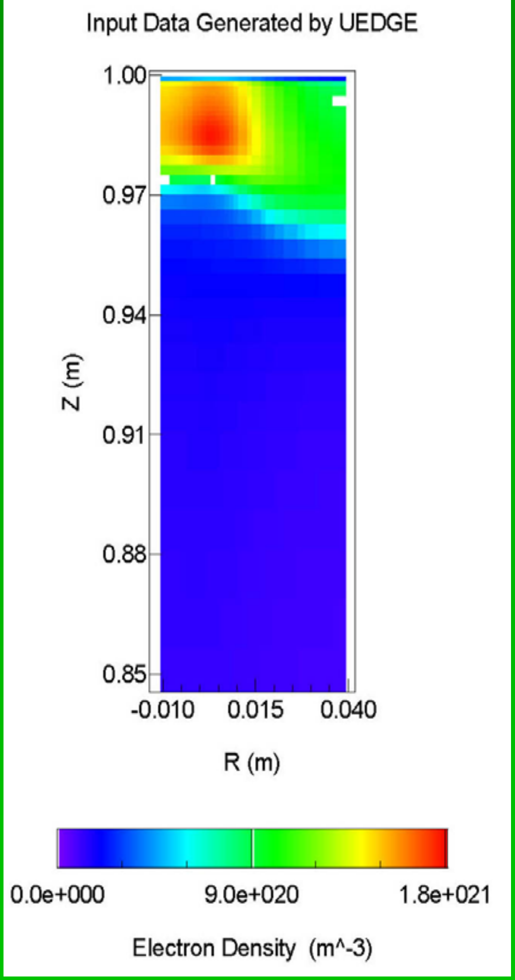
Analytic Fluid Neutral Model

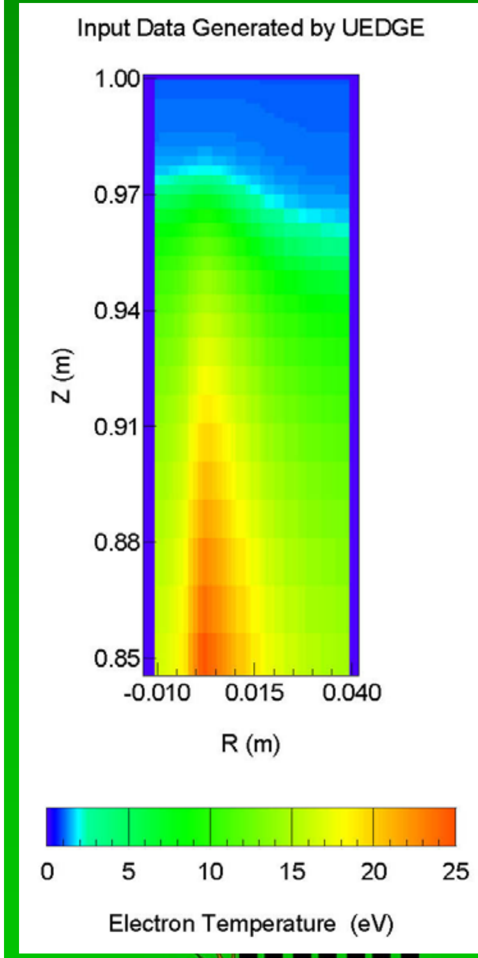
- 1-D fluid neutral momentum balance with constant CX cross section, linearly increasing T_i [Stotler 97] (courtesy S. Krasheninnikov),
- ~40 cases with DEGAS & DEGAS 2,
- Conclude:
  - Max(mfp/system size) < 0.1 for fluid model to be valid,
  - Min(mfp/grid spacing) > 0.5 to resolve thermal force.
- Code runs fine even if these are not true,
  - But, problem being simulated differs physically from the one postulated!



Code-Code Comparisons
DEGAS 2 – EIRENE

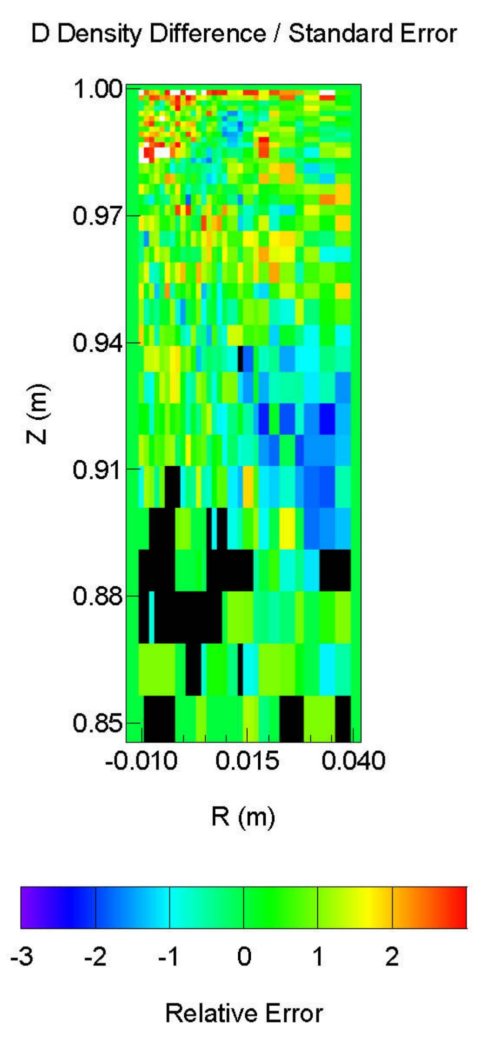
- Details in User's Manual (36 pp.).
- Codes share same basic algorithm & some data,
  - Yet, this was a lengthy exercise.
  - Similar story for DEGAS,
  - Different for KN1D.
- Addressed subtleties in handling of atomic & surface physics, scoring, & interpolation,
  - Required knowledge of internal workings of both.





Code-Code Comparisons DEGAS 2 – EIRENE

- Compiled histograms of differences normalized to Max(relative standard deviation, systematic error),
  - Compare histogram with expected fractions to estimate systematic error,
  - Conclude that systematic error is ~5%.
- Also did performance benchmark,
  - $\Rightarrow$ Initial optimization of DEGAS 2.



Solution Verification
Ideas for Going Further

- Establish database of verification test cases,
- Quantify accuracy of results [Oberkampf 02],
  - I.e., error trends in convergence studies,
  - Due to spatial and / or temporal discretization,
  - Iterative error,
  - Roundoff error.
  - Estimates will be useful in validation work!
- Analytic solutions
  - Undoubtedly many already exist & have been used,
  - Identify some for addition to test case database.
- Method of manufactured solutions [Roache 02],
  - Concoct solution that will exercise part or all of algorithm,
  - Analytically compute sources & boundary conditions associated with that solution,
  - Use these to set up run of code.
  - Solution need not be physical!
  - Useful solutions can be added to the test case database.
- Run metadata: record of details pertaining to particular run,
  - Code version,
  - Platform & compiler,
  - Location of input & output data files,
  - Else?

Outline

I. Introduction

II. DEGAS 2 Verification

  A. Code Verification

  B. Solution Verification

III. DEGAS 2 Validation

IV. Dedicated Validation Experiments

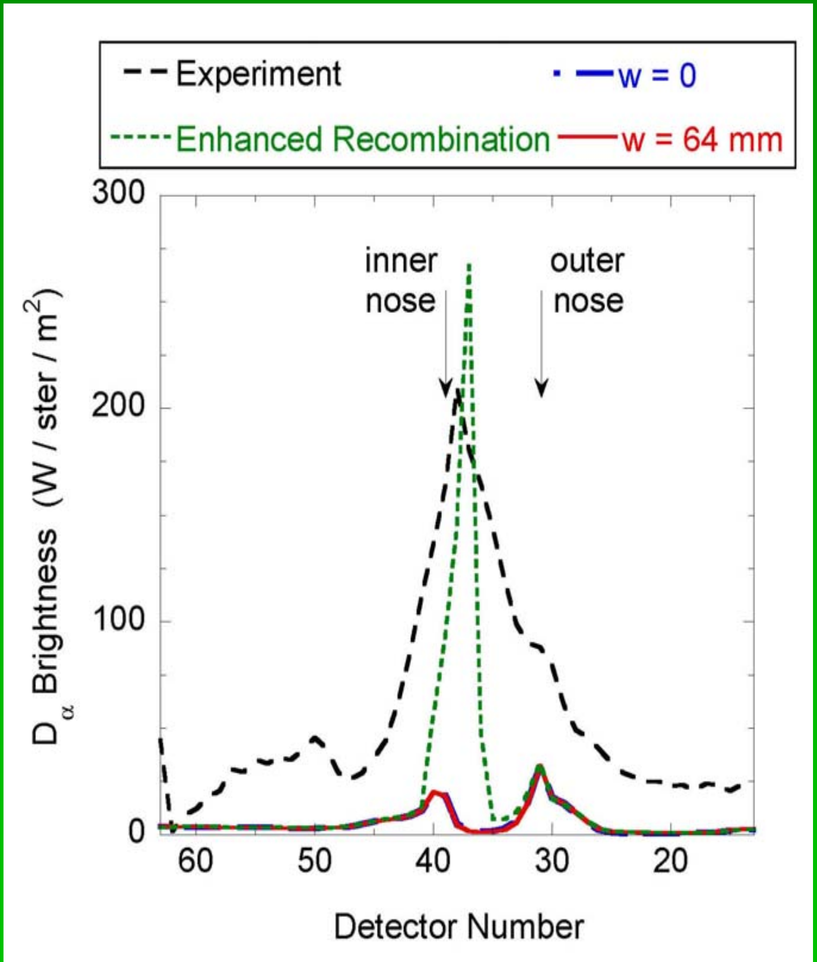
DEGAS 2 Validation History

1. TFTR H_α Spectrum in D-T [Stotler 96]
2. Alcator C-Mod Divertor Baffling Experiments
3. NSTX Gas Puff Imaging Experiments
4. C-Mod Gas Conductance Measurements

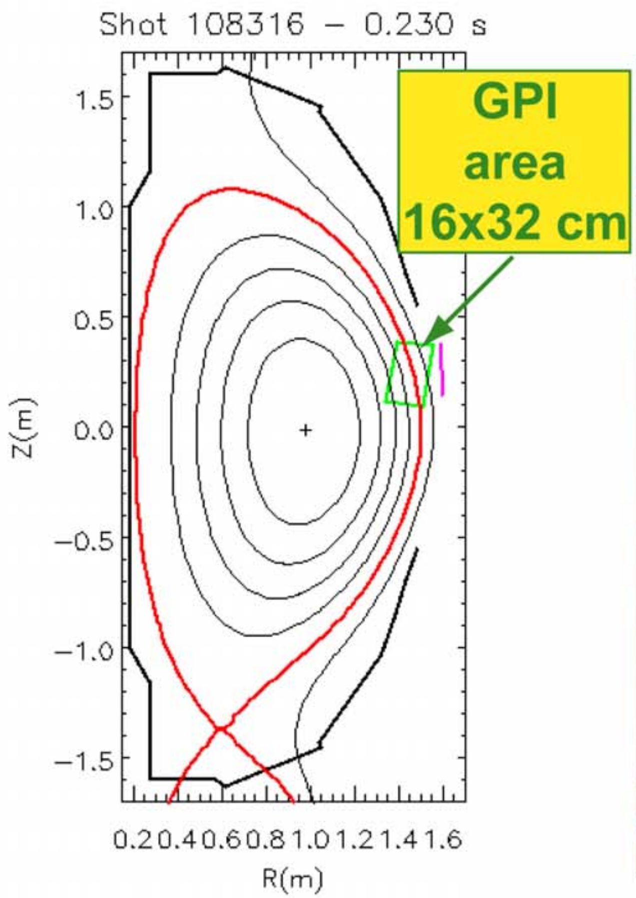
- None is an ideal validation in which code is run “blind”,
  - All code runs done after experiment,
  - Some code input based on experimental data.
  - Error analysis could be more rigorous.
- But, trend over the 4 cases is towards that ideal,
  - Not a conscious effort!
  - Physics content progressively simpler.
- All cases involved close interaction with experimentalists,
  - Acquire needed details about machine & experiment,
  - Understand subtleties & limitations of diagnostics.

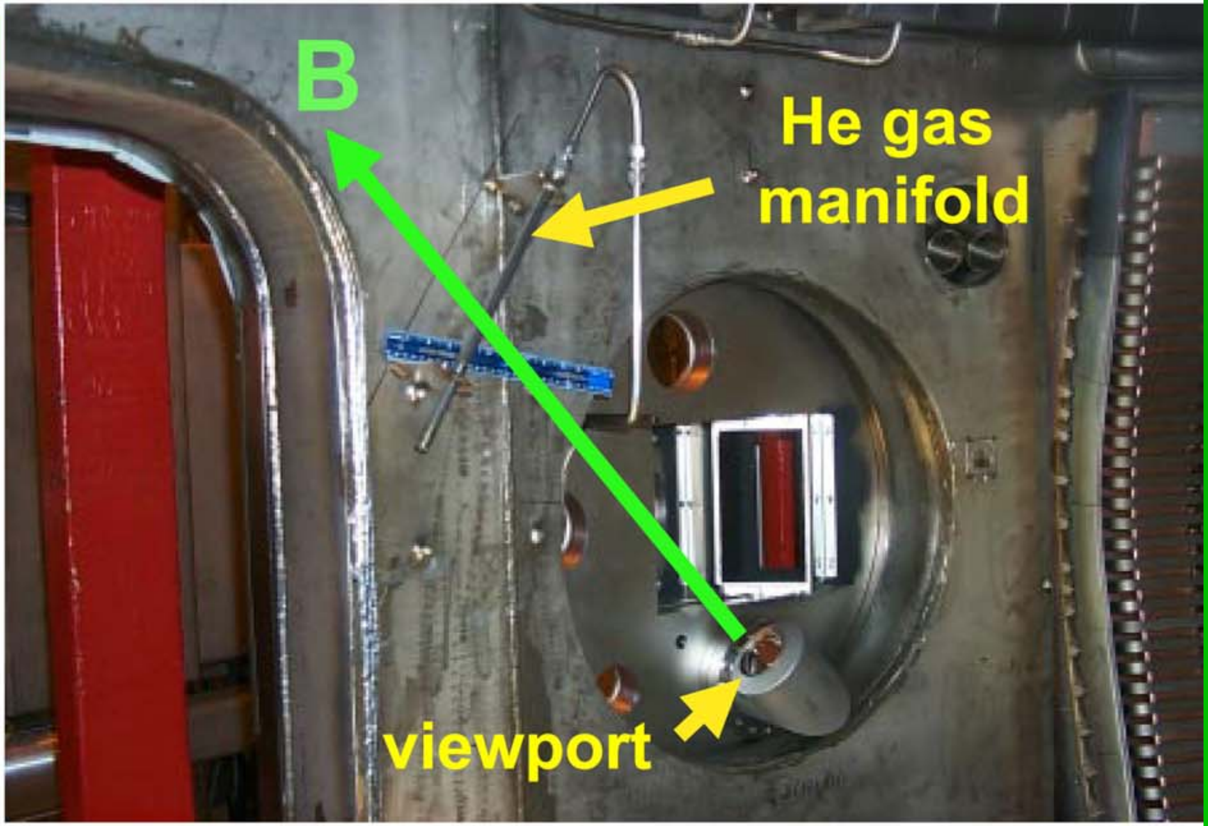
Alcator C-Mod Divertor Baffling Experiments [Stotler 01]

- Like TFTR case, simulate complex system,
  - Input to code again was experimental plasma data + model to fill in gaps,
  - Used 2-point model ⇒ no adjustable parameters,
  - Assumed divertor neutral pressure dominated by recycling on outer target.
- Found neutral pressures too low by ~10,
  - Divertor D_α also off.
  - Concluded that assumed plasma model was inadequate.
- [Lisgo 05] has found that PFR recombination dominates both,
  - Divertor pressure still off by 2 & error bars do not overlap.
  - Photon trapping & 3-D effects matter,
  - Other phenomena may enter
  - ⇒ not done adding physics!
- We did learn something,
  - But, did this increase our confidence in the code?



NSTX Gas Puff Imaging Experiments [Zweben 03, poster PI-18]





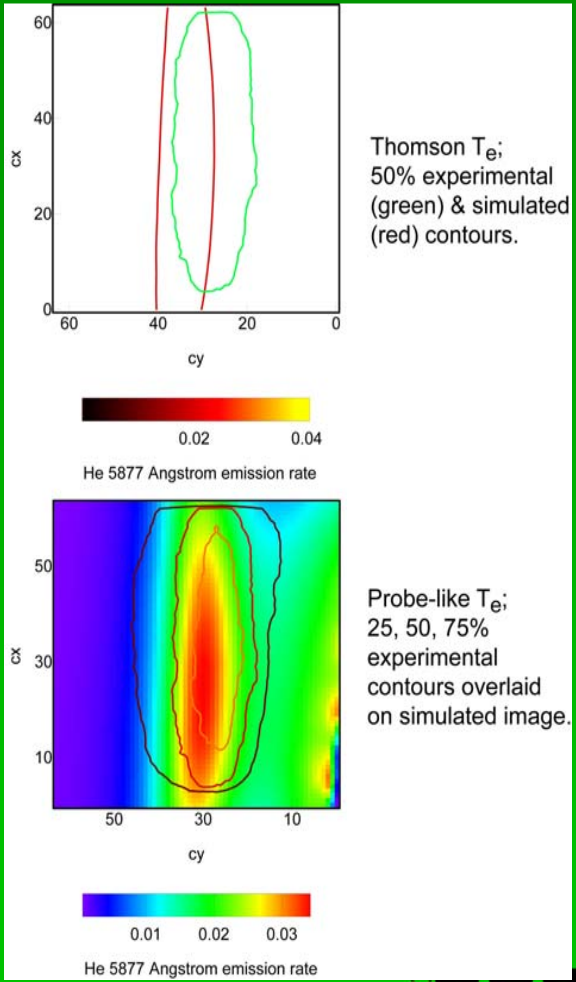
PPPL PRINCETON PLASMA PHYSICS LABORATORY

NSTX Gas Puff Imaging Experiments [Stotler 04]

- Our standard list of objectives for DEGAS 2 modeling:
  1. Demonstrate that we understand physics underlying GPI ⇔ validation,
  2. Use 3-D model to clarify spatial relationships of physical objects,
  3. Provide neutral density profiles that can be used to unfold 2-D n_e, T_e from camera images. [Myra, Edge & Pedestal I]
- Still need data from experiment as input:
  - Equilibrium,
  - Plasma n_e, T_e radial profiles.
- But, only "model" needed for plasma parameters is that they are constant on flux surface,
  - GPI experiment poloidally close to Thomson & probe locations.
- Simulated physics also simpler:
  - Only neutral source is gas puff,
  - Plasma recycling negligible,
  - Material interactions not significant.
  - Focused only on neutral transport & resulting light emission due to electron excitation,
  - Use of He as puffed gas reduces required atomic physics.
- Another example of direct simulation of diagnostic, the 64x64 pixel camera view,
  - Able to compare directly with experimental images,
  - Yielded 2-D, effective neutral density data used by Lodestar.

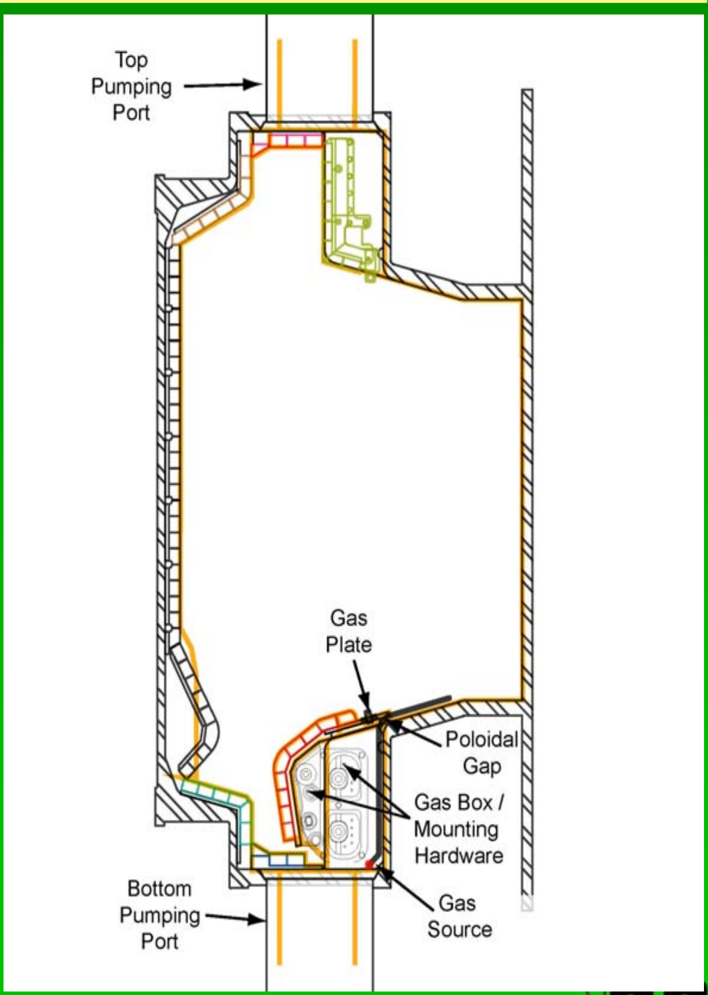
Could This Be A “Slam Dunk”?

- 1st complication: doing steady-state simulation, with constant on flux surface plasma, of time-varying, 3-D plasma,
  - Only have 1 or 2 TS profiles near GPI time.
  - Assume TS profile ↔ equilibrium,
  - & Compare with “median” experimental image computed from entire movie.
- 2nd complication: far SOL probe $T_e \gg$ TS T_e,
  - Discrepancy not surprising given large blobs there.
- Within error bars (not shown), simulated & experimental image peak locations agree.
- But, to really understand would need to account for impact of 3-D & time variation on TS & probe data as well as on emission cloud!



C-Mod Gas Conductance Measurements [LaBombard 03]

- Experiments were specifically designed to answer questions raised in modeling of divertor baffling experiments,
  - I.e., to validate the simulation codes!
- Focus here on measurements made without plasma,
  - Two cases: gas puff in open & closed divertor.
- ⇒ isolate physics components
  - Neutral viscosity,
  - 3-D representation of geometry.



Outline

I. Introduction

II. DEGAS 2 Verification

A. Code Verification

B. Solution Verification

III. DEGAS 2 Validation

IV. Dedicated Validation Experiments

Dedicated Validation Experiments

- "…validation experiments are indeed different from traditional experiments." [Oberkampf 02],
  - Primary distinction is that the code & code developer are primary customers.
- Not incompatible with present modes of operation,
  - But, making real progress with validation will require shift in priorities.
- Revisit some guidelines in [Oberkampf 04]:
  - Should be designed jointly by experimentalists & simulation scientists,
  - Experiment should measure all quantities needed as input to code,
  - Exploit synergies between code & experiment,
    - E.g., design experiment to offset a code weakness,
    - Use exploratory code runs to identify needed measurements.
  - Keep experimental & code results independent, i.e., blind comparison.
- [Oberkampf 02] goes on to discuss validation metrics,
  - Involving experimental & simulation uncertainties,
  - Postpone consideration for now.

Opportunities for Validation Experiments

- Comparing simulation results against tokamak discharges represents the top level of the validation hierarchy,
  - If they don't match, how do you know where the model is going wrong?
  - Developing confidence in model requires going through entire hierarchy,
  - Focus here on lowest level(s).
- My experience (this talk): even simplest benchmarks or experimental comparisons are worthwhile,
  - Rarely as simple as they seemed,
  - Always learn something.
  - Even demonstrating that physics is different than in full system constitutes progress.
- Plasma turbulence codes may be good place to start,
  - At or close to "first principles" $\Rightarrow$ no adjustable parameters,
  - Universal character of edge turbulence [Zweben 04] $\Rightarrow$ conceivably could make significant progress w/o tokamak experiments.
- Dedicated validation experiments ideal for university environment,
  - Relatively small scale compatible with typical university funding,
  - Multiple diagnostics $\Rightarrow$ opportunities for student involvement.
  - Requisite connection with code developers & theorists would keep university researchers & primary fusion labs closely coupled.
- Ideas for possible validation experiments?
  - Will need expertise of both experimentalists & simulation scientists,
  - TTF is perfect forum for this!

Conclusions / Discussion

- Code verification
  - Tools for improving software quality,
  - Static deep-flow analysis,
  - Dynamic testing.
- Solution verification
  - Database of accurate test cases from analytic results & MMS.
  - Code-code comparisons: may be tough!
  - Quantify accuracy of results.
  - Run metadata?
- Code Validation
  - Simulating full tokamak behavior may provide limited insight,
  - More constrained experiments can still be challenging.
- Dedicated Validation Experiments
  - Code is the primary customer!
  - Can we design experiments for lowest tier of hierarchy?
- See also the V&V talks in the parallel sessions
  - Core Working Group II Thursday 10:10 – 11:30
  - Core Working Group V Friday 3:00 – 4:15

References

- **[Hatton 97]** L. Hatton, “The T Experiments: Errors in Scientific Software”, *IEEE Comput. Sci. Eng.* **4**, 27 (1997).
- **[LaBombard 03]** B. LaBombard and C. J. Boswell, “In-Situ Gas Conductance and Flow Measurements Through Alcator C-Mod Divertor Structures With and Without Plasma Present”, MIT PSFC Tech. Report PSFC/RR-03-6 (2003).
- **[Lisgo 05]** S. Lisgo et al., J. Nucl. Mater. **337-339**, 139 (2005).
- **[Oberkampf 02]** W. L. Oberkampf and T. G. Trucano, “Verification and Validation in Computational Fluid Dynamics”, *Prog. in Aero. Sci.* **38**, 209 (2002).
- **[Oberkampf 04]** W. L. Oberkampf, “Verification and Validation in Computational Simulation”, Transport Task Force Meeting, Salt Lake City, Utah (April 2004).
- **[Post 05]** D. E. Post and L. G. Votta, “Computational Science Demands a New Paradigm”, *Physics Today* **58**, 35 (January 2005).
- **[Roache 98]** P. J. Roache, *Verification and Validation in Computational Science and Engineering* (Hermosa Publishers,Albuquerque, 1998).
- **[Roache 02]** P. J. Roache, “Code Verification by the Method of Manufactured Solutions”, *ASME J. Fluids Eng.* **124**, 4 (2002).

References (cont.)

- **[Stotler 94]** D. P. Stotler and C. F. F. Karney, "Neutral Gas Transport Modeling with DEGAS 2", *Contrib. Plasma Phys.* **34**, 392 (1994). See also `http://w3.pppl.gov/degas2`
- **[Stotler 96]** D. P. Stotler et al., "Modeling of Neutral Hydrogen Velocities in the Tokamak Fusion Test Reactor", *Phys. Plasmas* **3**, 4084 (1996).
- **[Stotler 97]** D. P. Stotler et al., "DEGAS 2 Neutral Transport Modeling of High Density, Low Temperature Plasmas", 16th International Conference on Plasma Physics and Controlled Nuclear Fusion Research (Montreal, Canada, October 1996) (IAEA, Vienna, Austria, 1997), Vol. **2**, p. 633.
- **[Stotler 01]** D. P. Stotler et al., "Modeling of Alcator C-Mod Divertor Baffling Experiments", *J. Nucl. Mater.* **290-293**, 967 (2001).
- **[Stotler 04]** D. P. Stotler et al., "Three-Dimensional Neutral Transport Simulations of Gas Puff Imaging Experiments", *Contrib. Plasma Phys.* **44**, 294 (2004).
- **[Stotler 05]** D. P. Stotler and B. LaBombard, "Three-Dimensional Simulation of Gas Conductance Measurement Experiments on Alcator C-Mod", *J. Nucl. Mater.* **337-339**, 510 (2005).
- **[Zweben 03]** S. J. Zweben et al., "High-Speed Imaging of Edge Turbulence in NSTX", *Nucl. Fusion* **44**, 134 (2004).

PPPL
PRINCETON PLASMA PHYSICS LABORATORY

Supplementary Slides

What is DEGAS 2?

- A Monte Carlo neutral transport code [Stotler 94]
  - Fast, efficient
    - Parallelization built in at the start
    - Most arrays allocated dynamically
  - Portable
    - Karney's parallel random number generator
    - Most data exchanges via netCDF files
  - Easy to add new physics
  - Written in FWEB,
    - Grungy tasks handled by macro preprocessor,
    - TeX documentation interwoven with code.
  - Statistics:
    - 31 executables:
      - Main code
      - 13 setup
      - 10 test
      - 7 post-processor
    - ~120 source files
    - Source code (including internal documentation): ~70K lines.
    - 85 atomic & surface physics data files

Does CVS Stand for “Crummy, Vexing Software”?

- Use CVS (Concurrent Version System) to maintain code versions,
  - Including binary data & output files!
  - “Tags” keep track of code releases & branches.
  - CVS has no “undo” $\Rightarrow$ be careful!

I Don't Have Time to Write Documentation!

- Write documentation as code is being developed,
  - Especially in source code,
    - Explain less-than-obvious code,
    - You'll be glad you did in ~3 years.
    - Be sure comments & source code agree!
  - Where applicable, refer to equations & expressions in published literature.
- Compile a User's Manual,
  - Writing comprehensive manual very time consuming,
  - Many sections of DEGAS 2 User's Manual were written first for posters or talks,
    - It is my reference of choice for these topics!
- Document reference data files,
  - E.g., DEGAS 2's atomic and plasma-surface interaction data files.
  - Have one file containing brief description of origin of each data file.
  - Atomic physics data generated by codes I run are also tracked by CVS.

```
if (Day.eq.'Thursday') then
   x = 2.
else
   x = 3.
end if
```

Implicit None: Use It, Love It

- Primary value is to catch misspelled variable names.
- Much easier to have compiler point them out to you than to track down an obscure bug at run time.
- Strong typing (e.g., Java) would be even better.

assert(cell_intersect < geom_infinity)

- An “assertion” checks conditions that ought to be true,
  - Print warning,
  - And / or stop code.
- If a line of code makes some assumption about a variable in it, check it! E.g.,
  - Array index $\leq$ dimension,
  - Denominator $\neq$ 0,
  - Particle is still in problem space.
- Very useful in catching memory corruption,
  - Memory corruption can make the “impossible” come true.
- In DEGAS 2, used in:
  - All setup, testing, and codes,
  - Some post-processing codes,
  - Initial runs of main program.
  - Disabled for production runs.

Let Computer Do Grungy Coding

- DEGAS 2 uses FWEB based macros to:
  - Handle dynamic memory allocation,
  - Specify common blocks, including with MPI,
  - Read & write netCDF files.
- E.g., the string `gi_ncread(fileid)`expands to ~280 lines of code I don't have to maintain.
- Represents "heroic" application of FWEB by Karney,
  - Not clear how to best do this in general.

Suns are From Venus, Linux Machines are From Mars

- Trying different platforms & compilers ensures portability. Duh!!!
  - Beneficial to do so before your users try it.
- Some compilers picky about syntax,
  - "Best" is NAG F95.
- Compilers / platforms differ in subtle ways,
  - E.g., handling of uninitialized variables,
  - $\Rightarrow$ Comparing runs on different systems will help find such bugs.
- For parallel codes, compare single & multiple processor runs.

Testing of DEGAS 2 Geometry

- Tracking flights through problem space is key component of MC algorithm,
  - Code must always know where flight is!
- Geometry problems very hard to catch at run time,
  - E.g., might be very small volume of phase space entered only after ~10^6 flights.
- $\Rightarrow$ Essential to test thoroughly during problem development.

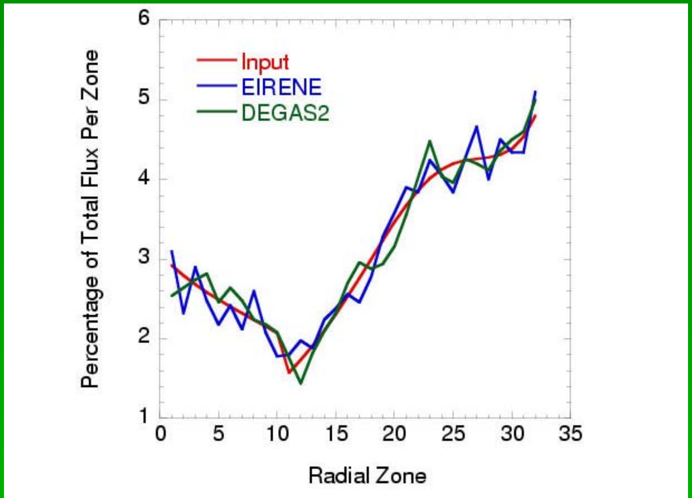
1. Basic consistency checks of geometry objects & connectivity information,
   - E.g., a surface with something on one side, but nothing on other must be “outer” surface.
2. Divide problem space into “pixels” & track through them,
   - Same routine used to generate 2-D & 3-D plots.

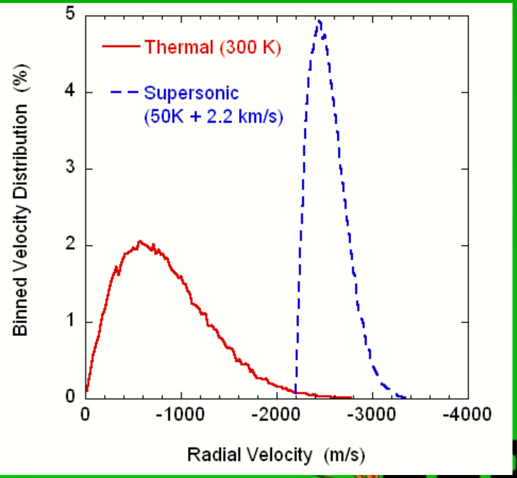
PPPL PRINCETON PLASMA PHYSICS LABORATORY

Testing of Physical & Velocity Space Sampling of DEGAS 2 Sources

- Know distributions & expected sampling error ⇒ verify!

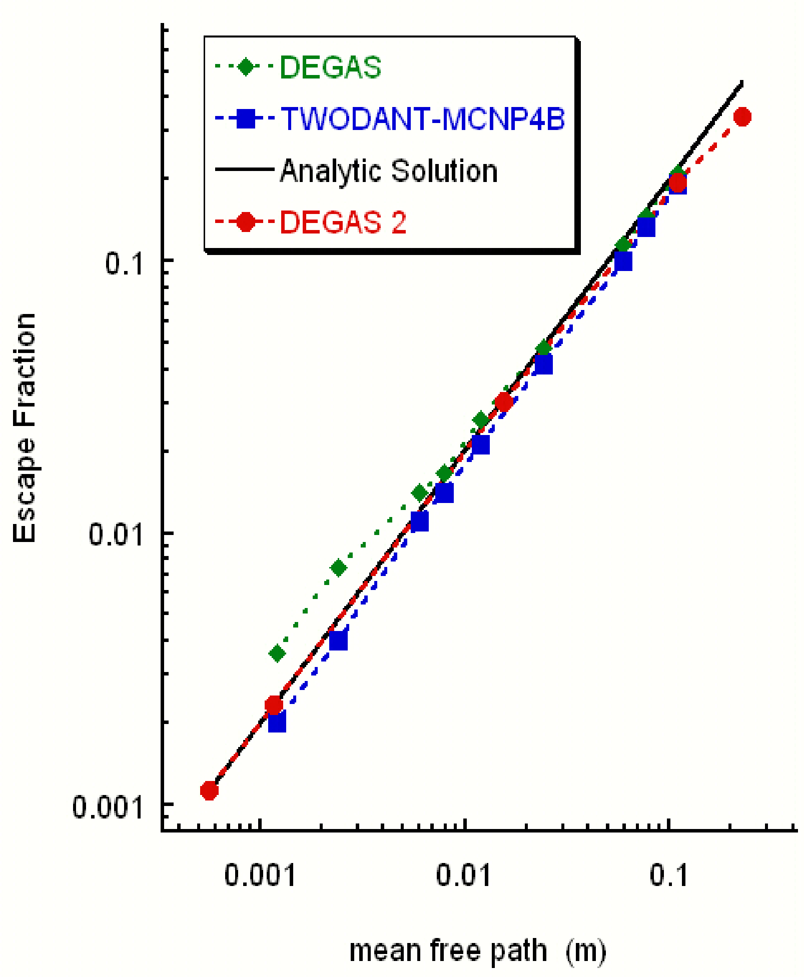
1. Compare sampled distribution of flights over source "segments" with requested distribution,
   - Check scaling of deviations with # flights,
   - Consistency quantified by χ^2 test.
2. Compare sampled spatial & velocity distribution of source at given segment with expectations,
   - Exact same random number sequence as main code,
   - Dump out each flight x, v as well as averages,
     - Load flight data into KaleidaGraph to check higher moments or make scatter plots.





Tests Against Analytic Solutions
Escape Probability

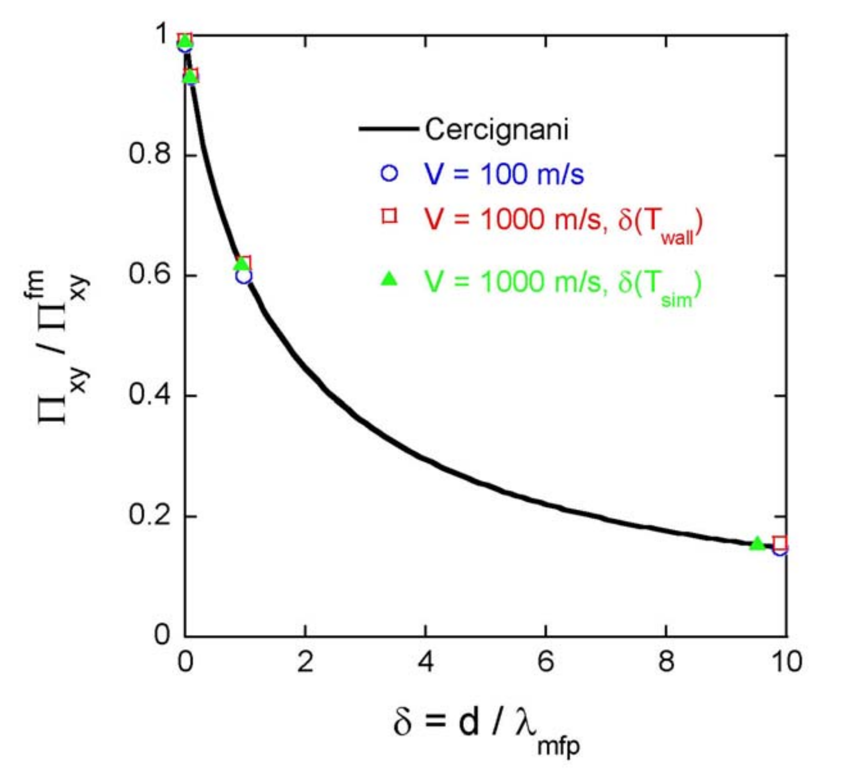
- Volume source in box ≫ ionization mfp (courtesy R. Rubilar, ca. 1998)
  - → mfp * Area / (4 * Volume).
- Old DEGAS had problems due to overlap between “external” & “tracking” meshes.
- Low mfp TWODANT & MCNP data lacked precision.
- What’s missing?
  - Error bars!
  - On analytic expression,
  - & on MC results.



PPPL PRINCETON PLASMA PHYSICS LABORATORY

Couette Flow
Test of Neutral Viscosity

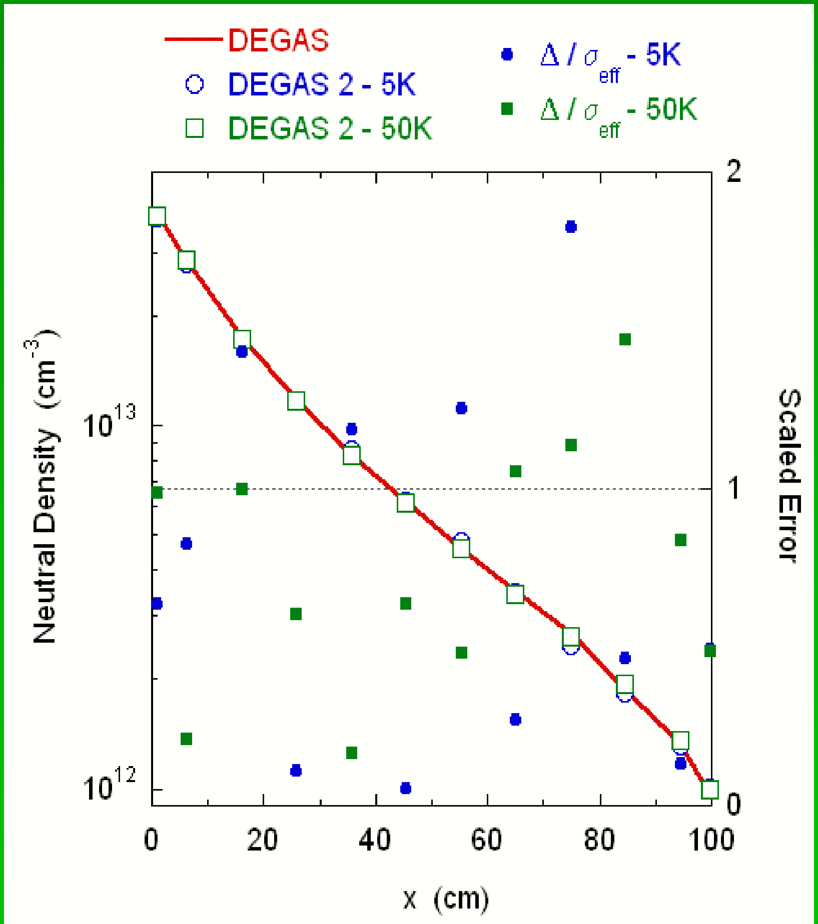
- Fluid flow between sliding plates, d apart, relative velocity V (courtesy D. Reiter, Chr. May).
- Getting free molecular (kinetic) limit right requires "Maxwell flux" distribution ⇒ flux leaving wall = flux hitting wall.
- Still missing error bars,
  - Now have statistical + iterative errors.
- Also have from Reiter & May, 1 & 2 component equilibration tests,
  - Time-dependent & not part of standard DEGAS 2 release.
- Don't be fooled by all this success!
  - BGK model used for neutral-neutral collisions known to get thermal conductivity wrong,
  - Not tested by these cases,
  - ⇒ be alert for situations where thermal conductivity matters!



PPPL PRINCETON PLASMA PHYSICS LABORATORY

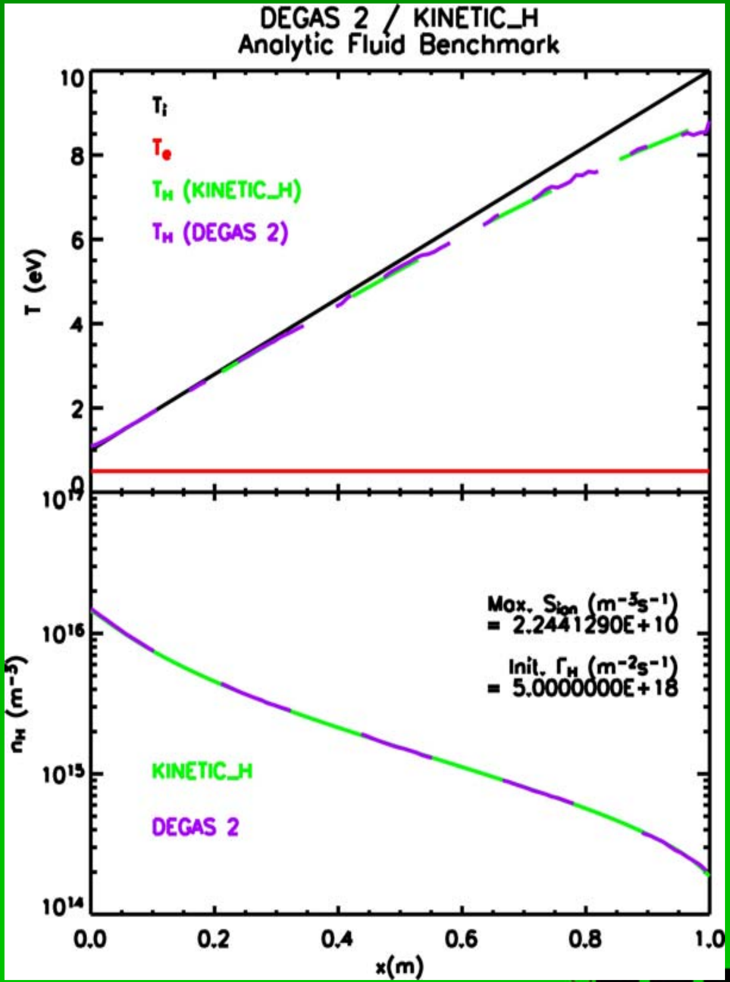
DEGAS 2 – DEGAS Comparison

- Based on variant of analytic fluid model run.
- Roughly same story as in EIRENE benchmark.
- Getting agreement required knowledge of internal operation of both codes:
  - Coordinating meshes,
  - Ensuring $\langle \sigma v \rangle_{CX}$ was same.
- Plot shows error scaled by standard deviation,
  - Relative standard deviations differ by ~2.
  - For uncorrelated errors, should have 68% < 1, 95% < 2.



DEGAS 2 – KN1D Comparison

- KN1D: 1-D continuum, kinetic code (B. LaBombard),
  - Fundamentally different numerical approach complicates resolving discrepancies.
  - Plus, types of possible code modifications different.
- Initial case: analytic fluid model,
  - Issues found included CX cross section, v-space resolution,
  - Able to get good match.
- Subsequent attempts at more realistic modeling (e.g., adding ionization) faltered,
  - In part due to time constraints.
- Bottom line on code-code comparisons: anticipate expending considerable resources,
  - Especially if codes based on different algorithms, have incompatible geometries, etc.

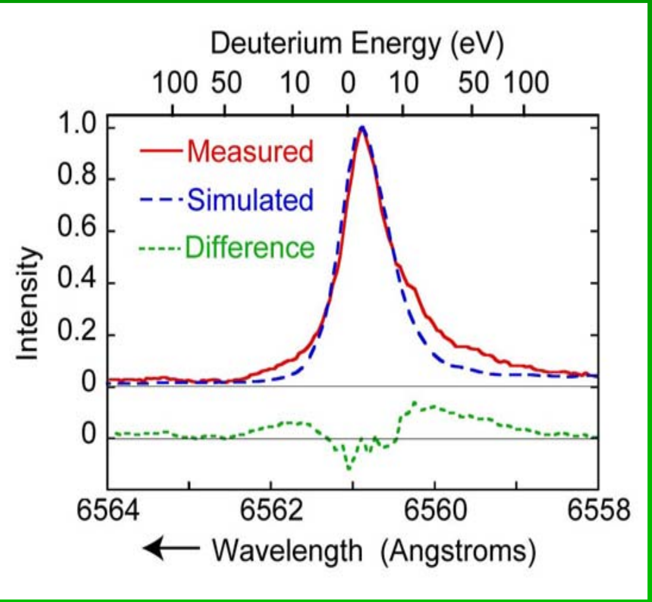


Conservation Checks

- A legitimate verification test,
- Level of difficulty ranges from trivial to great depending on algorithm,
  - E.g., in DEGAS 2, conservation is exact with “collision” scoring, but statistical with “track length”.
  - Former is still worth doing: used energy conservation to demonstrate that original implementation of BGK collisions was incorrect.
- Related: symmetry tests,
  - E.g., If problem & boundary conditions possess some symmetry, does solution?

TFTR H_α Spectrum in DT [Stotler 96]

- Actually done with DEGAS, but instructive,
- H_α spectrum ⇔ neutral velocity distribution (synthetic diagnostic),
- Plasma n, T & ion fluxes input to DEGAS obtained from TRANSP + SOL plasma model fit using H_α chord data,
  - Core data had error bars,
  - Did some sensitivity analyses,
  - But, did no comprehensive error analysis.
- Inferred by process of elimination that spectrum difference due to high energy dissociative recombination of H_2^+,
  - Concluded that we needed better data on H_2^+, including vibrational excitation.
- Very complicated system!
  - Did this really increase confidence in the code?



PPPL PRINCETON PLASMA PHYSICS LABORATORY

DEGAS 2 Simulations of C-Mod Gas Conductance Experiments [Stotler 05]

- **Find:**
  - Closed: U_{exp} = 1.2 m^3/s, U_{sim} = 0.83 $\pm$ 0.06 m^3/s,
  - Open: U_{exp} = 4.5 m^3/s, U_{sim} = 7.8 $\pm$ 1.2 m^3/s.
  - In opposite direction! $\Rightarrow$ Single fix unlikely!
- **Simulation error (closed case):**
  - MC noise + iterative variation ~ 0.01 m^3/s,
  - Uncertainty in geometry details: -0.07 m^3/s,
  - Change in U_{sim} with increased spatial resolution = 0.05 m^3/s.
    - Issue became apparent following simulations of fluid flow in 3-D square pipe.
- **Experimental errors:**
  - Similar pressures measured at multiple locations with different gauge types $\Rightarrow$ error likely small.
  - One other parameter: vessel volume = 4.06 m^3,
    - Uncertainty unknown,
    - Estimate based on simulated volume $\simeq$ 3.2 m^3.