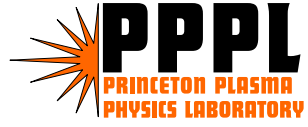
GRIN

GReen's function INtegrals

Irek Szczesniak
and
Alexander Pletzer

Princeton Plasma Physics Laboratory
September 2001



PART I

Green's function method and the GRIN program

- Green's function method
- Examples
- The GRIN program

Green's Function Method

The Green's function is the response to a Dirac excitation.

Suppose we want to solve:

$$\boldsymbol{\nabla}^2\psi \ = -4\pi\rho(\vec{x})$$

and we know the Green's function G satisfying:

$$\nabla^2 G(\vec{x'},\vec{x}) = -4\pi\delta(\vec{x}-\vec{x'})$$

which corresponds to the potential due to a point charge.

Then the solution is:

$$\psi(\vec{x'}) = \int_V G(\vec{x'},\vec{x})\rho(\vec{x})dV$$

Green's Second Identity

Central to the Green's function method is Green's second identity

$$\oint_S d\vec{S} \cdot \{G(\vec{x'}, \vec{x})\nabla\psi(\vec{x}) - \nabla G(\vec{x'}, \vec{x})\psi(\vec{x})\} - 4\pi\alpha\psi(\vec{x'}) = -4\pi \int_V dV G(\vec{x'}, \vec{x})\rho(\vec{x})$$

which is obtained after multiplying $\nabla^2\psi = -4\pi\rho(\vec{x})$ by G, integrating over the volume V, and using the definition of G.

Here,

$$\text{where } \alpha = \begin{cases} 1 & \text{if } \vec{x'} \in V \\ \frac{1}{2} & \text{if } \vec{x'} \in S \\ 0 & \text{if } \vec{x'} \notin V \cup S \end{cases}$$

Boundary Conditions

The intuitive solution (sum of all source contributions)

$$\psi(\vec{x'}) = \int_V G(\vec{x}, \vec{x'})\rho(\vec{x})dV$$

is a special case of the general solution

$$\oint_S d\vec{S} \cdot \{G\nabla(\vec{x'}, \vec{x})\psi(\vec{x}) - \nabla G(\vec{x'}, \vec{x})\psi(\vec{x})\} - 4\pi\alpha\psi(\vec{x'}) = -4\pi \int_V dV G(\vec{x'}, \vec{x})\rho(\vec{x})$$

which also takes into account the effect of inhomogeneous boundary conditions.

First Example

Given:

$$G(x, y, x', y') = -log((x - x')^2 + (y - y')^2)$$

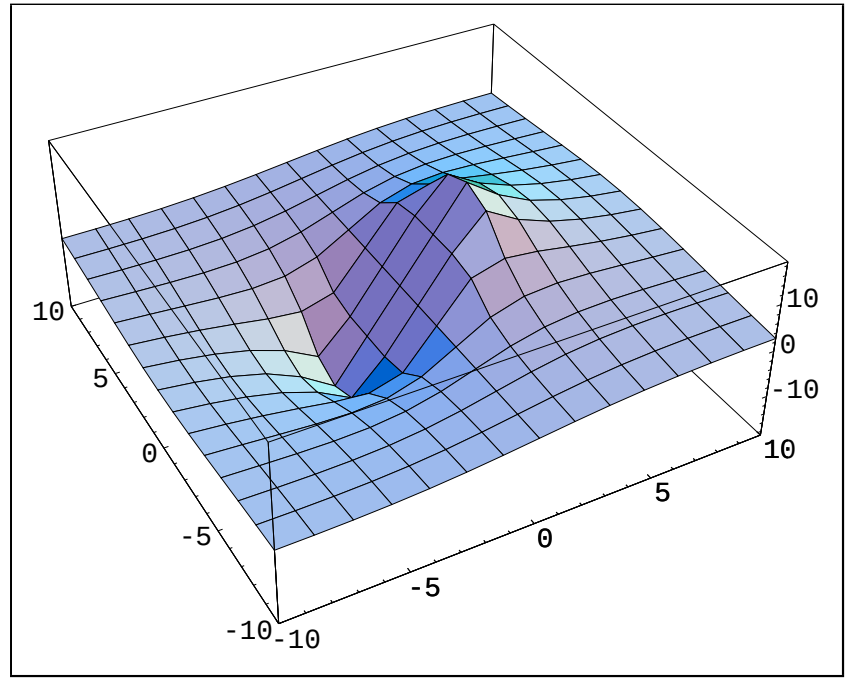
Find the potential ψ due to a source $s(t) = cos(t)$ distributed on a ring:

$$\begin{cases} x = 3cos(t) \\ y = 3sin(t) \end{cases}, \quad t \in (0, 2\pi)$$

Solution:

$$y(x', y') = \int_0^{2\pi} G[x', y', x(t), y(t)]s(t)\sqrt{\frac{dx(t)}{dt}^2 + \frac{dy(t)}{dt}^2}\, dt$$

Solution to the First Example



The figure depicts potential ψ.

$$\psi \sim r \text{ inside}$$

$$\psi \sim \frac{1}{r} \text{ outside}$$

Second Example

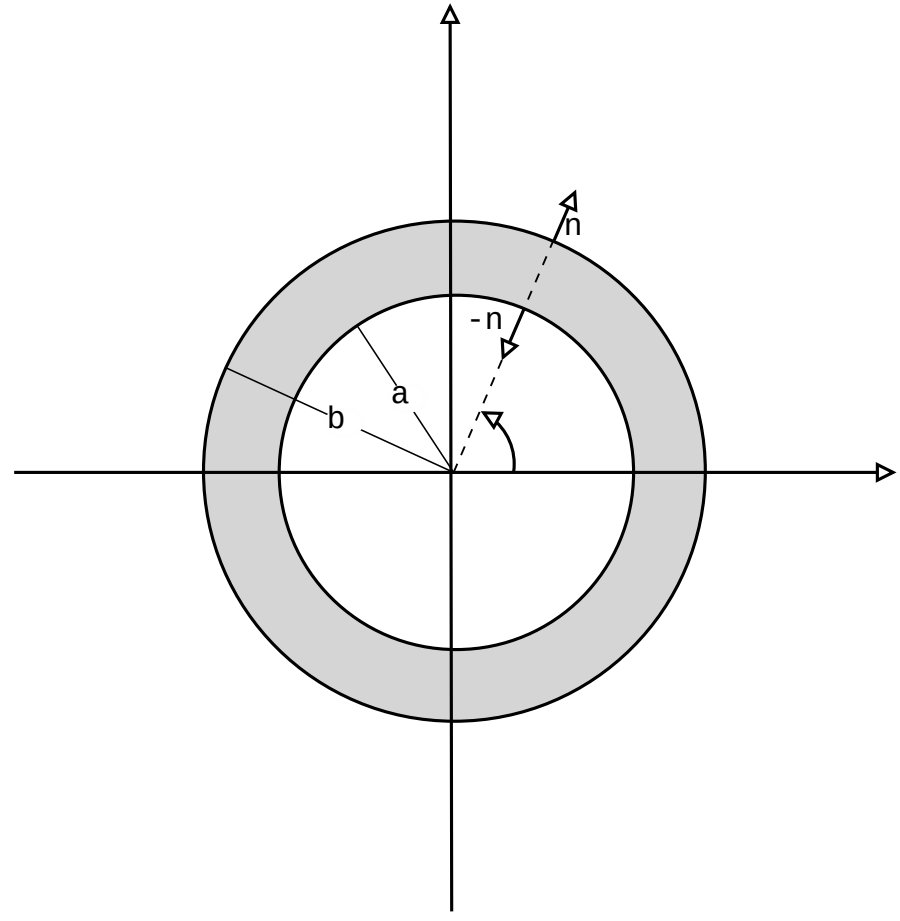
Given:

$$\psi(\theta) = \begin{cases} U(\theta) & \text{for } r = b \\ 0 & \text{for } r = a \end{cases}$$

Solve:

$$\nabla^2 \psi = 0$$

in the shaded region
$(a < r < b)$
to find the *normal*
electric field E on $r = a$.



Second Example (continued)

Green's second identity for the example reads:

$$\int_b bd\theta\{-G(\vec{x'},\theta)\vec{E}_b(\theta)-\frac{\partial G}{\partial n}U(\theta)\}-\int_a ad\theta\{-G(\vec{x'},\theta)\vec{E}_a(\theta)+\frac{\partial G}{\partial n}0\}-4\pi\alpha\psi(\vec{x'})=0$$

For $\vec{x'}$ on a:

$$\int_b bd\theta G(\vec{x_a}(\theta'),\theta)E_b(\theta)+\int_b bd\theta\frac{\partial G(\vec{x_a}(\theta'),\theta)}{\partial n}U(\theta)=\int_a ad\theta G(\vec{x_a}(\theta'),\theta)E_a(\theta)$$

For $\vec{x'}$ on b:

$$\int_b bd\theta G(\vec{x_b}(\theta'),\theta)E_b(\theta)+\int_b bd\theta\{\frac{\partial G(\vec{x_b}(\theta'),\theta)}{\partial n}-\frac{2\pi}{b}\delta(\theta'-\theta)\}U(\theta)=$$

$$\int_a ad\theta G(\vec{x_b}(\theta'),\theta)E_a(\theta)$$

Second Example (continued)

Expand

$$E_a = \sum_i E_a^{(i)} e_i(\theta)$$

$$E_b = \sum_i E_b^{(i)} e_i(\theta)$$

$$U = \sum_i U^{(i)} e_i(\theta)$$

in basis functions $e_i(\theta)$. We then get a coupled linear system of equations:

$$\mathcal{A} \cdot \mathbf{E_b} + \mathcal{B} \cdot \mathbf{U} = \mathcal{C} \cdot \mathbf{E_a}$$

$$\mathcal{D} \cdot \mathbf{E_b} + \mathcal{E} \cdot \mathbf{U} = \mathcal{A} \cdot \mathbf{E_a}$$

After elimination of $\mathbf{E_b}$, a linear relation between $\mathbf{E_a}$ and $\mathbf{U}$ can be obtained, and the problem is solved.

Punch Line

To solve the above type of problems, we need the capability to:

- compute *accurately* line integrals involving a kernel with a log singularity,
- solve dense linear system of equations.

GRIN

The GRIN code computes integrals of two kinds:

$$\int_C K(l',l)\rho(l)dl$$

$$\int_{C'} \nu(l') \int_C K(l',l)\rho(l)dldl'$$

with $K(l',l) \sim log(l'-l)$ as $l' \to l$ if $C = C'$ (true for elliptic operators in 2-D).

GRIN vs. VACUUM

VACUUM is a highly Successful code written by M. Chance, which is used to compute the natural boundary conditions in a number of stability codes (PEST, DCON, GATO, ...).

Feature	VACUUM	GRIN
portability	CRAY, alpha[a]	UNIX
max. # of contours	hardcoded	hardware limited
geometry of contours	hardcoded	almost arbitrary
choice of kernel	1 (hardcoded)	arbitrary with log singularity
basis function	Fourier, finite elements	user supplied

[a] ongoing work to port to other Unixes

Green's functions provided by GRIN

There are presently 10 kernel functions (G or $\frac{\partial G}{\partial n}$) that come with GRIN. An example is the toroidally averaged Green's function for the Laplace equation (following Chance [*Phys. Plasmas* **4**, 2161 (1997)]):

$$G_n(R,Z;R'Z') = \left(\frac{1}{\pi R'R}\right)^{\frac{1}{2}} \Gamma(\tfrac{1}{2}+n)\, p^{\frac{n}{2}+\frac{1}{4}}\, \frac{F\left(n+\frac{1}{2},\frac{1}{2},n+1|p\right)}{\Gamma(n+1)}$$

$$p \equiv \frac{s-1}{s+1}, \qquad s \equiv \frac{\lambda}{\sqrt{\lambda^2-1}}, \qquad \lambda \equiv 1+\frac{(R'-R)^2+(Z'-Z)^2}{2R'R}$$

where $F(a,b,c|p)$ is the *Gauss hypergeometric function.*

More that comes with GRIN

GRIN has vector and matrix classes implemented.

The matrix class is interfaced to Lapack.

C++ example of matrix operations:

```
Mat Temp = MatMult(kmat_ab, Inverse(kmat_bb));

Vec chi_a = MatMult(
                    MatMult(
                            Inverse(kmat_aa - MatMult(Temp, kmat_ba)),
                            gmat_aa - MatMult(Temp, gmat_ba)
                    ), dchin_a);
```

Example of GRIN usage

Define the geometry (segments)

```
/* Define segments */

Vec tb(11), ta(5);
segment sa, sb;
ta.space(0., 1.);
tb.space(0., TwoPi);
Vec xa(5); xa = 2.;
Vec ya(5); ya.space(-0.1, 0.1);
sa.load(ta, xa, xb, not_periodic);
sb.load(tb, xb, yb, periodic);
```

Example of GRIN usage (continued)

Compute $\int d\ell' K(\ell, \ell')\alpha'(\ell')$

```
/* set observer point */

double tobs = 0.5;

/* call procedure */

/* greenLaplaceCartesian = Green's fct
   alpha is the basis fct (ext proc) */

double res = greenIntegral(
     greenLaplaceCartesian,
     sa, tobs, sb, alpha);
```

Example of GRIN usage (continued)

Compute $\int_s d\ell \alpha(\ell) \int_{s'} d\ell' K(\ell, \ell') \alpha'(\ell')$

```
/* beta, alpha are the basis fcts */

double res = twoGreenIntegral(
     greenLaplaceCartesian,
     sb, beta, sa, alpha);
```

GRIN's NICHE

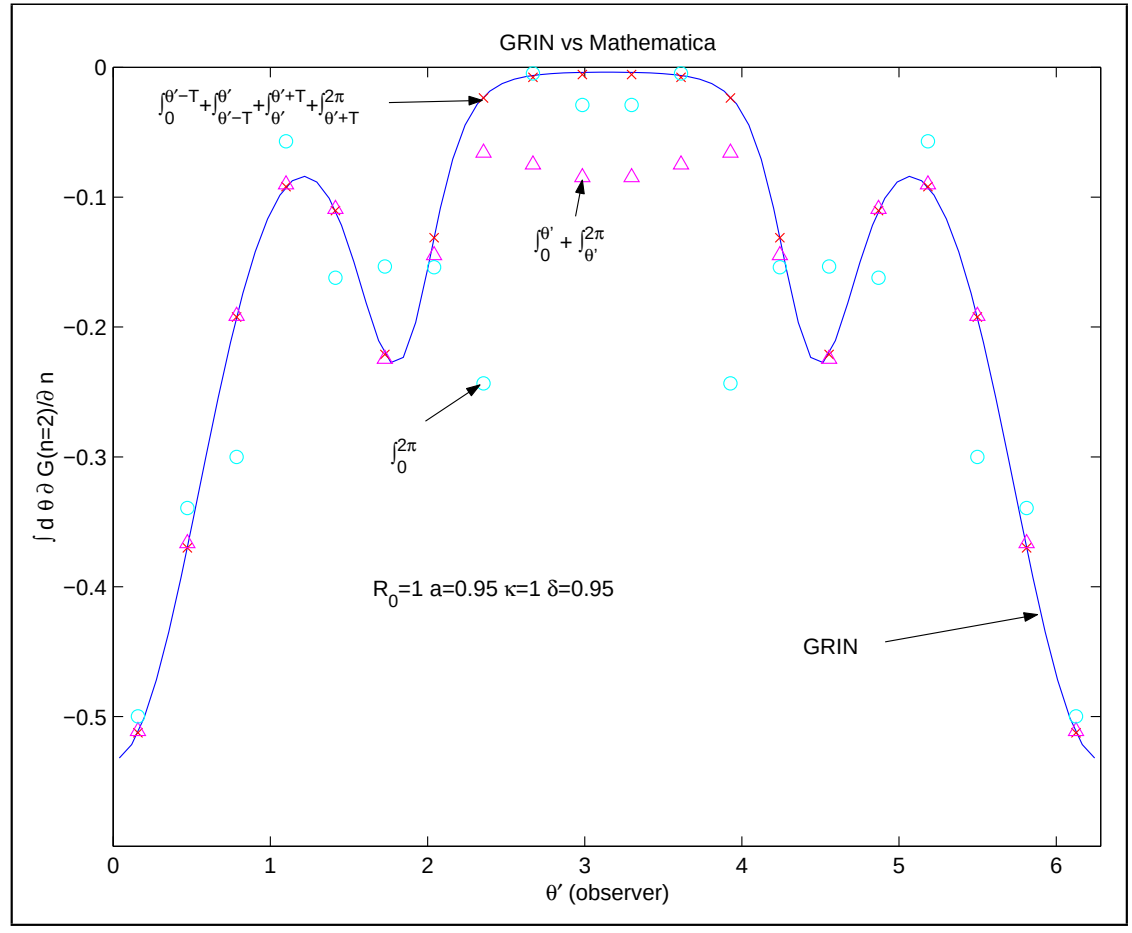
Reasons to develop GRIN

There is a gap between general software as Mathematica and specialized software as VACUUM. This gap is filled in by GRIN.

Feature	Mathematica 3	GRIN	VACUUM
Accuracy	depends	very good	very good
Flexibility	very good	good	poor
Performance	poor	good	very good
Friendliness	very good	good	poor
Programming	from scratch	tools provided	hard to change

Inaccurate Results of Mathematica 3

Reasons to develop GRIN



Discrepancy between GRIN's and Mathematica 3 results.

Inaccurate Results of Mathematica 3 (continued)

The circles represent the value of $\int_0^{2\pi} d\theta K(n=3)$ returned by the NIntegrate Mathematica function for various observer coordinates.

The triangles are those returned by NIntegrate when splitting the interval in two. The crosses (x) are those values returned by NIntegrate after further splitting; these agree much better with the GRIN results obtained using greenIntegral.

GRIN's results are more accurate because of the proper handling of the log singularity.

PART II

Improvements made to GRIN

- Accuracy and performance: Green's functions and ${}_2F_1$
- Portability: building the code on different platforms
- Robustness: rely on extensively tested software when possible

Accuracy and Performance

The Green's functions for the toroidal Laplace equation are expressed in terms of the Gauss hypergeometric series (${}_2F_1$). As the series converges slowly for some arguments, it is prone to inaccuracy and bad performance.

The ${}_2F_1$ Function

The ${}_2F_1$ function is represented by the Gauss hypergeometric series:

$$F(a,b,c|x) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!}$$

where:

$$(a)_k = \underbrace{a(a+1)...(a+k-1)}_{\text{k terms}}, (a)_0 = 1$$

The series has the convergence radius $r = 1$ in the complex plane. For $x \in (0,1)$ the series converges. For $x >= 1$ the series diverges. For $x \to 1^-$ the series converges slowly.

The ${}_2F_1$ Function

The most important Green's function (the toroidally averaged Green's function for the Laplace equation) uses ${}_2F_1$ in the following form:

$$F(n+\tfrac{1}{2},\tfrac{1}{2},n+1|x) \text{ for } x \in (0,1)$$

This can be computed using the series:

$$F(a,b,a+b|x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(k!)^2} [2\psi(k+1) -$$

$$\psi(a+k) - \psi(b+k) - \psi(1-x)](1-x)^k$$

$$\text{for } x \in (0,1)$$

which converges very well for $x \to 1^-$, but slowly for $x \to 0^+$.

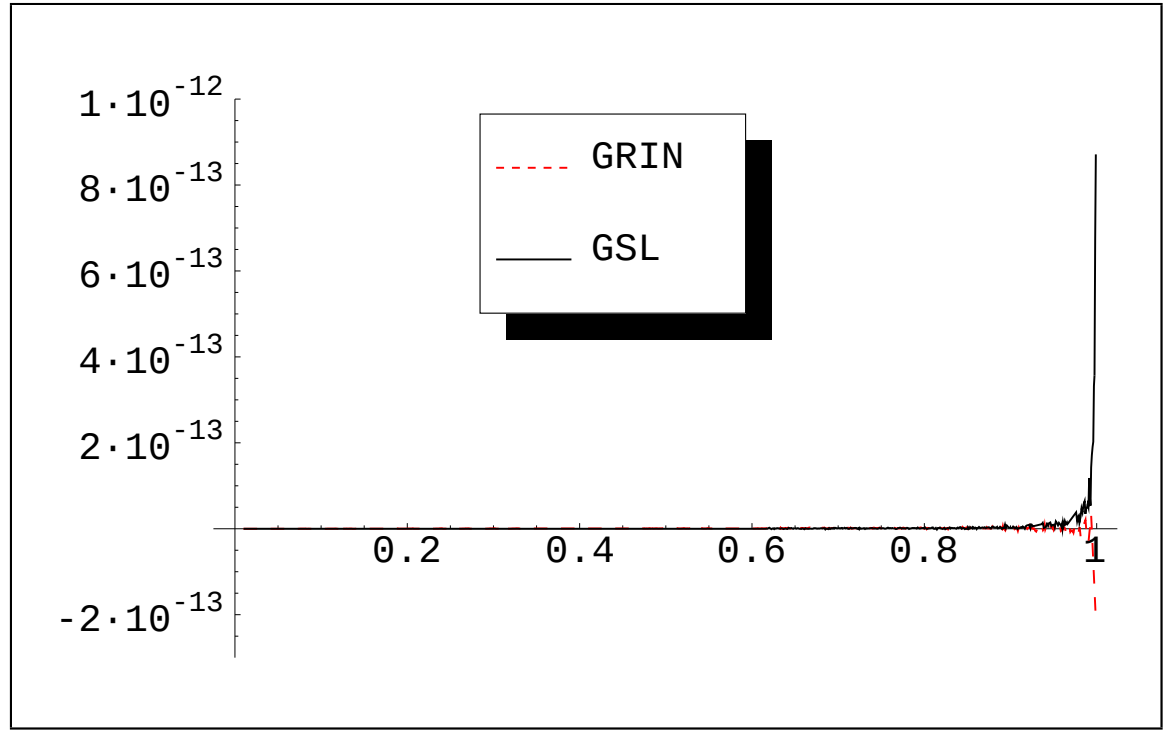
Use ${}_2F_1$ from GSL?

GNU Scientific Library (GSL) provides only one general function (`gsl_sf_hyperg_2F1`) to compute:

$$F(a, b, c|x) \text{ for } x \in (0, 1)$$

Accuracy Comparison of GRIN's and GSL's ${}_2F_1$

GSL's ${}_2F_1$ inaccurate near x = 1;

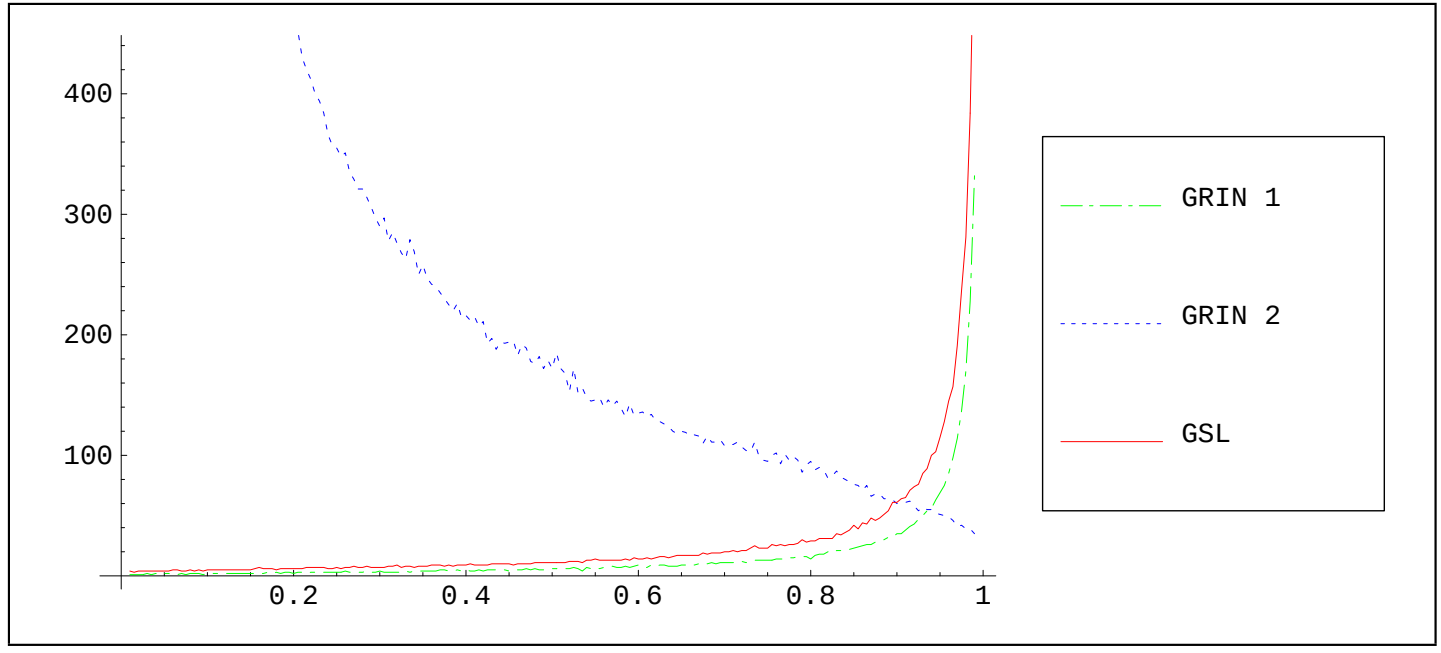


Absolute error of GRIN's and GSL's ${}_2F_1$.

GRIN's ${}_2F_1$ returns results for $0 \leqslant x < 0.9999999999999999$.

GSL's ${}_2F_1$ returns results for $0 \leqslant x < 0.999$ only.

Speed Comparison of GRIN's and GSL's ${}_2F_1$



Time consumption of GRIN's and GSL's ${}_2F_1$.

The ordinates represent the time in milliseconds consumed by calculation of $F(n+\frac{1}{2},\frac{1}{2},n+1|x)$ done 10000 times for the given x and $n=1$. The timing was on a Sun SUNW, Ultra-4 computer.

"GRIN 1" uses standard series, "GRIN 2" uses the asymptotic series.

Robustness and Portability

GRIN uses standard tools that have been developed and tested over many years. These tools guarantee better portability and better reliability of GRIN.

Dependencies:

- Autoconf,
- STL,
- LAPACK.

Autoconf

Autoconf is a GNU tool for user unattended configuration of software packages that makes software portability easier.

The tool has been used and developed over ten years. Every GNU package employs *Autoconf*.

Autoconf is capable of guessing:

- platform type,
- compilers,
- libraries,
- dependencies.

STL

Standard Template Library

STL is a standard C++ library available with every C++ compiler that conforms to the C++ ANSI standard.

The library offers data structures such as vectors, sets or queues. Additionally there are algorithms such as sorting, binary searching or finding set intersection.

LAPACK

Linear Algebra PACKage

LAPACK is a library for solving matrix equations:

- systems of linear equations,
- least-squares solutions of linear systems of equations,
- eigenvalue problems,
- and singular value problems.

LAPACK is a widely used standard library for numerical computation.

New *Vector* and *Matrix* Classes

GRIN was written at the time when not every C++ compiler supported STL.

Feature	MV++	STL
used by	old classes	new classes
performance	very good	very good
portability	not guaranteed	all platforms
documentation	exists	very good
widely used?	no	very

GRIN vs. VACUUM

Test case: Compute the vacuum energy matrix

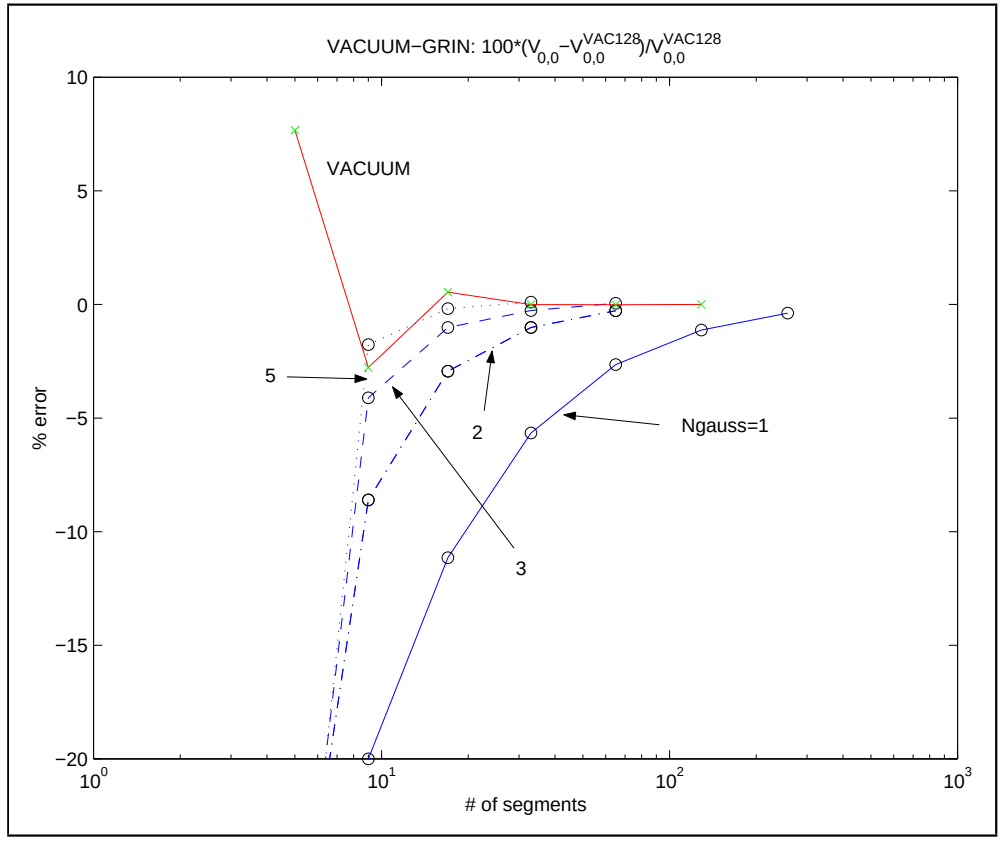
$$(m - nq)V_{mm'}(m' - nq)$$

in the magnetohydrodynamic stability codes PEST, DCON, etc., for a plasma of inverse aspect ratio $\epsilon = a/R = 0.9/1$, elongation $\kappa = 2$, triangularity $\delta = 0.9$,

$$\left.\begin{aligned} x &= R + a\cos(\theta + \delta\sin\theta) \\ y &= \kappa\sin\theta \end{aligned}\right\}$$

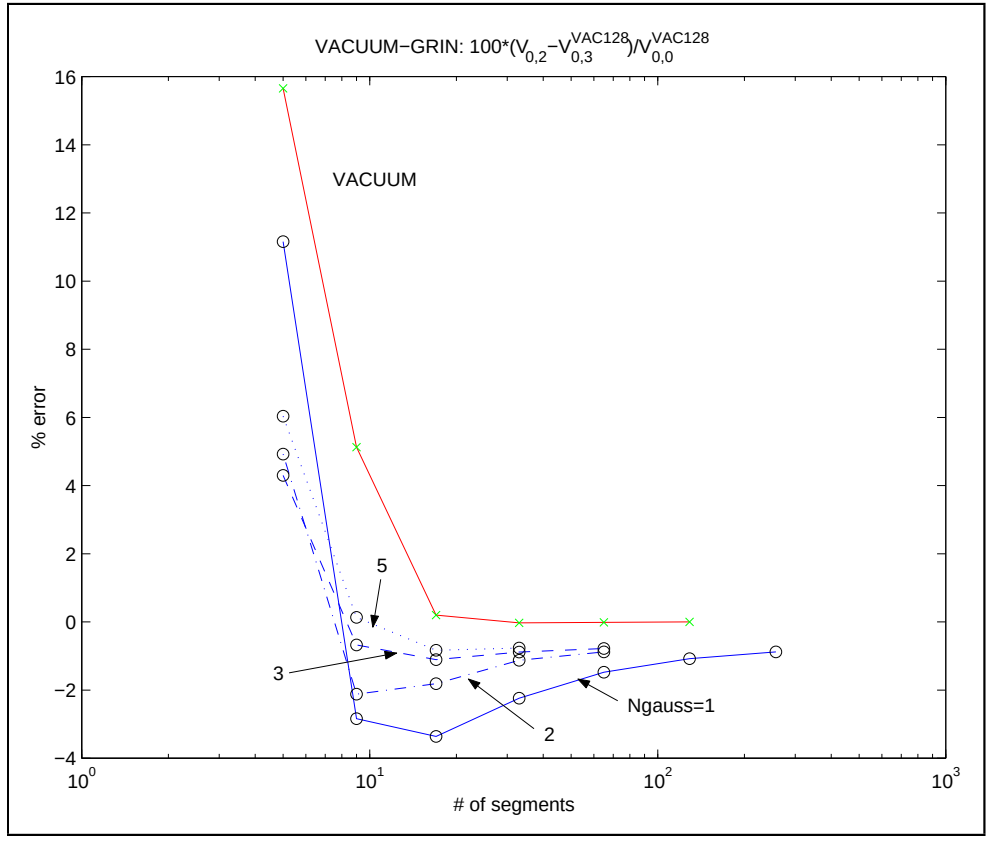
The toroidal mode number is $n = 1$ and the poloidal mode numbers vary from $-2 \cdots + 2$. Domain extends from infinity to plasma boundary (no wall).

GRIN vs. VACUUM (continued)



Normalized difference of V_{00} computed by GRIN (blue) and VACUUM. Both codes agree in the limit of increased resolution and quadrature order (`ngauss`=number of Gauss points in each segment).

GRIN vs. VACUUM (continued)



Difference of V_{03} normalized to V_{00}. GRIN and VACUUM give a slightly different value.

Future Work

Work that *can* be done:

- improvements to the Gauss quadrature,
- parallel approach with MPI,
- introduce Automake.

Conclusions

After months of development GRIN is:

- fully working and ready to solve real problems,
- well tested (passed 10 test cases, which included various geometries tests and internal mechanisms tests),
- flexible (a user specifies the problem: geometry, Green's function, basis functions).

GRIN will become available at http://w3.pppl.gov/NTCC in October 2001.