J-Solver Equilibrium Code

Alex Pletzer
Jon Menard
Princeton Plasma Physics Laboratory

October 19, 2000

1 Introduction

J-Solver is a fixed-boundary, toroidal MHD equilibrium code written at the Princeton Plasma Physics Laboratory by Steve Jardin, Jon Menard, C. Kessel, D. Monticello, with contributions from A. Pletzer and others [1]. The code solves the Grad-Shafranov equation in flux coordinates (ψ, θ) and returns two arrays X and Z as functions of the flux ψ and the equal-arc poloidal angle θ. Various other quantities including the safety factor, the poloidal flux, the pressure profiles and metric quantities are also calculated and are returned upon request.

The Grad-Shafranov equation has two free functions of the poloidal flux, which uniquely specify the toroidal current. The Grad-Shafranov equation is then solved by iterating the cylindrical X and Z coordinates until the error falls below a specified tolerance. The mesh iteration is constrained by the requirement that the coordinate system has equal arc-length in the poloidal angle. One important feature of J-Solver is that the mesh can be doubled after a selected number of iterations to improve both the convergence and the accuracy of the solution.

The name J-Solver derives from the fact the flux-averaged current density parallel to the magnetic field is prescribed, the other free function being the pressure gradient. Specifying the parallel current instead of the safety factor profile, for instance, allows a straightforward incorporation of various current drive sources into the Grad-Shafranov equation such as steady state ohmic, neoclassical bootstrap, and external current drive.

2 Running J-Solver

J-Solver requires the plasma geometry and two input profiles: $p'(\psi)$ and $\frac{<\vec{J}\cdot\vec{B}>}{<\vec{B}\cdot\nabla\phi>}$. These profiles can be specified in several ways. Perhaps the most intuitive way is to represent these profiles as sets of sampling data points in flux coordinates. To ensure backward compatibility, J-Solver can also read the inequ file, where the geometry and profiles are parameterized. Alternatively, the IFUNC2=4 switch in inequ allows to interface the output profiles generated by the Tokamak Simulation code (TSC) with J-Solver.

2.1 User specified geometry and profiles

This mode of operation is particularly suited for embedding J-Solver into a larger program. A call to J-Solver follows at least three steps: the first ensures that enough memory is allocated, the second consists in running the core J-Solver and the third ensures that the previously allocated memory is freed.

```
      CALL jsoInit(MN, NDOUB, ISYM) ! initialize and
                                    ! allocate memory

      ! ... (list of optional ‘‘set’’ routines)

      CALL jsoExec(
     & FACIMP,
     & GZEROS,
     & KMAX, NPSIT,
     & PPRIME_S, AJB,
     & XBND, ZBND,
     & PSIBAR0,
     & X, Z,
     & IERROR)

      ! ... (list of optional ‘‘get’’ routines)

      CALL jsoFree ! reclaim memory
```

where

- MN -(input) Determines the initial grid size to be $2^{MN}+1$ poloidally, and 2^{MN} radially.
- NDOUB -(input) Number of times the grid is doubled. After $NDOUB$ mesh doublings have occurred, the grid size becomes $2^{MN+NDOUB}+1$ times $2^{MN+NDOUB}$.
- ISYM -(input) Plasma up-down symmetry switch: 0 (asymmetric) or 1 (symmetric).
- FACIMP -(input) Error tolerance for inner loop elliptic equation solver. If FACIMP is negative, | FACIMP | will be used as the error tolerance, and will be divided by 5 after each grid doubling.
- GZEROS -(input) Toroidal field normalization at the plasma edge ($\vec{B}\cdot\nabla\phi/|\nabla\phi|^2$).
- KMAX -(input) Number of independent plasma-vacuum boundary points. The total number of boundary points is $KMAX+1$.
- NPSIT -(input) Size of input profile arrays.
- PPRIME_S -(input) Pressure gradient profile $dp/d\hat{\psi}$, where $\hat{\psi}\equiv\psi/\psi_{max}$.
- AJB -(input) Parallel current density array $\frac{<\vec{J}\cdot\vec{B}>}{<\vec{B}\cdot\nabla\phi>}$.
- XBND -(input) Radial coordinate of the plasma-vacuum boundary of size $\geq KMAX+1$. Due to periodicity, the last node XBND(KMAX+1) must coincide with XBND(1).
- ZBND -(input) Vertical coordinate of the plasma-vacuum boundary of size $\geq KMAX+1$. The last node ZBND(KMAX+1) must coincide with ZBND(1).
- PSIBAR0 -(input/output) Total poloidal flux enclosed by the plasma/2π.
- X -(input/output) Radial coordinate of the plasma mesh returned as a one-dimensional array of size NTHEP*NPSI, where

  $NTHEP=(ISYM-1)*(2**(MN+NDOUB)+ISYM)+1-2*ISYM$

  and

  $NPSI=2**(MN+NDOUB)+1$.

Neighboring poloidal points are contiguous in memory. (The values of $NTHEP$ and $NPSI$ can be obtained using a "get" call, see below.)

- Z -(input/output) Vertical coordinate of the plasma mesh returned as a one-dimensional array of size NTHEP*NPSI. As for X, neighboring poloidal points are contiguous in memory.

- IERROR -(output) Integer error flag. A value of 0 indicates successful completion.

  Between initialization and memory clean-up, all quantities defined in the f90 common-modules can be accessed and modified through "get" and "set" routine calls, respectively. The list of admissible "set" calls includes all the parameters that are otherwise accessible in the inequ file plus some additional ones. For instance

  ```
  CALL jsoSetNimax(11)
  ```

  resets the number of inner-loops before the metric is updated.

  All these input parameters take default values that need not be changed for most applications. Thus the "set" routines allows the users to concentrate on those parameters that are critical to their simulations while ensuring that the default parameters are not inadvertently changed.

  Similarly there are "get" routines to access internal variables. At present, only a limited set of these calls are implemented; users are encouraged to write their own access routines to suit to their needs. In the following example,

  ```
  CALL jsoGetPp(PPRIME)
  ```

  the pressure gradient profile $\frac{dp}{d\psi}$ is accessed. The size of PPRIME must be NPSI.

2.1.1 Warm start capability

Many applications involve running an equilibrium code many times while slightly varying some parameter. It is appropriate in this case to use the warm-start capability to accelerate the numerical convergence. A warm-start can be achieved by setting NDOUB to a negative value (e.g. -1) in the initialization call,

```
CALL jsoInit(MN, -1, ISYM)
```

In this case the input values of PSIBAR0, X and Z are used as "first guess." There is no mesh doubling associated with warm-starts. The sizes of X and Z arrays are therefore $NTHEP * NPSI$ and must be pre-allocated, with
$NTHEP = (ISYM - 1) * (2 * *MN + ISYM) + 1 - 2 * ISYM$
and
$NPSI = 2 * *MN + 1$.

3 Using the inequ input file

The inequ file has up to recently J-Solver's traditional input file. A single call

```
CALL jsoInequ(ierror)
```

suffices in this case to initialize, perform the equilibrium computation and reclaim the memory. A detailed description of the inequ file in given in Sect. 7.

3.1 Interfacing with the Tokamak Simulation Code (TSC)

TSC produces a file eqdska. This file can be converted to a NetCDF file eqdsa.cdf using the tsc2cdf utility. The eqdsa.cdf can then read by J-Solver after setting IFUNC2=4 in inequ.

Note that some TSC versions dump 8-byte long integers into the eqdska file. Long integers are not portable. If you suspect this to be the case try using the tsclong2cdf utility on a platform that supports 8-byte long integers. This produces a eqdsa.cdf file that is fully portable.

4 Ouput files

By default, J-Solver produces three files after a successful run: eqdsk.cdf, jso.profiles and jso.scalars. Two small database files dbequp and dbequs can in addition be produced by setting DBWRT = 1 in inequ (or use the corresponding "set" call).

The files jso.profiles and jso.scalars contain information about global parameters and profiles in ASCII format. By far the largest file, eqdsk.cdf is a collection of radial profiles (pressure, safety factor etc.) and (X, Z)

arrays, which serves as interface to a collection of PPPL codes such as PEST, NOVA, DMAP etc. The eqdsk.cdf file is written in NetCDF format (http://www.unidata.ucar.edu/packages/netcdf/) using the EZCDF routines—a set of easy-to-use interface routines available from http://w3.pppl.gov/NTCC. The content of eqdsk.cdf can be viewed by typing

```
ncdump eqdsk.cdf | more
```

An example showing how to read eqdsk.cdf data is provided in the reqdsk.f file.

J-Solver can run silently by setting

```
CALL jsoSetIverbose(0)
CALL jsoSETIeqdsk(0)
```

5 Major changes in this version

Major code changes have been undertaken to make this J-Solver version portable across UNIX platforms (LINUX PCs, workstations and CRAYs), dynamically allocatable, user friendly and avoid the use of proprietary software. J-Solver has been converted to fortran 90 syntax and makes use of IMPLICIT NONE statements. Dimensions are computed on the fly, which simplifies the user interface. It also greatly reduces the average memory requirement. In the process, however, some previously available features have been lost or abandoned:

- NCAR graphics. All calls to NCAR graphics have been commented out but can be switched on by executing the commands (in Bash)

  ```
  for f in `ls *.f`; do
  cat $f | perl -ne 's/ctv80\/ncarg//o; print;' > foo
  mv $f $f_old
  mv foo $f
  done
  ```

  Rather than mingling graphics with number crunching code, the preferred approach is to custom write routines that dump data into a file for postprocessing. Examples of such routines are provided in the drive.F file.

- Restart option from dump file (IRST1=1 or IRST2=1) removed. This feature has been superseded by the warm-start capability.
- Radial and poloidal grid sizes have been merged. Previous versions allowed to specify independent radial and poloidal grid sizes through the exponent MNTHE and MNPSI. The grid size is now uniquely defined by MN = MAX(MNTHE, MNPSI), this to overcome a bug which caused the code to crash when MNTHE differed from MNPSI.
- Previous versions of J-Solver produced the fort.17 file. This file is now replaced by eqdsk.cdf file, which is NetCDF portable.

6 Additional tools

Interpolation onto an arbitrary mesh can be achieved using interface routines to the PSPLINE library developed by D. McCune.

For a set of one-dimensional arrays (S, F) of sizes N that need to be interpolated using cubic splines onto a grid SI of size NI, the interpolation call is:

```
CALL jsoInterp1d(N, S, F, NI, SI, FI)
```

where FI is the output set of interpolated data at the SIs. Similarly there is a two-dimensional interpolation routine,

```
CALL jsoInterp2d(NT, NS, T, S, F, NTI, NSI, TI, SI, FI)
```

to interpolate an NT times NS array F — a function of NT poloidal grid points T and NS radial surface indexed by S — onto a new grid (SI, TI) of size NTI times NSI. The routine returns FI, an NTI times NSI matrix. Periodic boundary conditions are assumed in T with neighboring poloidal points contiguous in memory.

7 Description of the input file inequ

Failures in J-Solver can often be traced back to bad input in the inequ file. No tabs are allowed in inequ. Each column is strictly ten characters wide, and left justification must be obeyed.

7.1 Card 0 - Equilibrium calculation switches

- IRST1 - (obsolete).
- IRST2 - (obsolete).
- NDOUB - Number of times number of grid zones is doubled = NDOUB
- IPSI - Radial coordinate switch.
  0.0 $\Rightarrow$ Radial coordinate is $\sqrt{\psi}$.
  1.0 $\Rightarrow$ Radial coordinate is ψ.
- IFUNC2 - Profile form switch for parallel current, p, n, and T. IFUNC2 is a 3 digit number of the form $i_3 i_2 i_1$. Note: $\hat{\psi} \equiv (\psi - \psi_{axis})/(\psi_{edge} - \psi_{axis})$

  Profile functional form switch:
  $i_3 = 0 \Rightarrow$ Profile $(\psi) \sim \text{weight}_1 \times (1 - \hat{\psi}^{\text{b1}})^{\text{a1}} + \text{weight}_2 \times (1 - \hat{\psi}^{\text{b2}})^{\text{a2}}$.
  Profile information is taken from cards 2, 3, 9, and 10.
  $i_3 = 1 \Rightarrow$ Profile $(\psi) \sim$ 3 region cubic spline representation.
  Profile information is taken from cards 16, 17, 18, and 19.

  Pressure profile switch:
  $i_2 = 0 \Rightarrow$ Central density determined by P0 and TZER0.
  n profile determined from p and T profiles.
  $i_2 = 1 \Rightarrow$ Central density determined by P0 and TZER0.
  p profile determined from n and T profiles.
  $i_2 = 2 \Rightarrow$ Central pressure determined by DZERO and TZER0.
  p profile determined from n and T profiles.

  Parallel current profile switch:
  $i_1 = 1 \Rightarrow$ User-specified parallel current function.
  $i_1 = 2 \Rightarrow$ High bootstrap-fraction model.
  $i_1 = 3 \Rightarrow$ Self-consistent ohmic model.

$i_1 = 4 \Rightarrow$ Read in profiles from TSC. Requires NetCDF eqdska input file.

If i_1=4 and IFUNC2 is negative, only read in plasma shape.

$i_1 = 5 \Rightarrow$ Self-consistent ohmic-bootstrap model.

If $i_1 = 3$ or 5, i_2 is set to 2 internally.

- IFILE - Number of records skipped in file EQDSKA when IFUNC2 = 4.

- NUMIT - Maximum number of metric outer interations before quitting.

7.2 Card 1 - Computation grid switches

- MNTHE - Relates to the number of poloidal zones, $2^{\mathrm{MNTHE}} + 1$.

- MNPSI - Relates to the number of ψ surfaces, $2^{\mathrm{MNPSI}}+1$. This version only implements a single size MN=MAX(MNTHE, MNPSI).

- ISYM - Plasma up-down symmetry switch.

  0.0 $\Rightarrow$ Up-down asymmetry.

  1.0 $\Rightarrow$ Up-down symmetry. ZZERO must = 0.

- ISPH - Special profile switch for Spheromaks.

  0.0 $\Rightarrow$ Tokamak.

  1.0 $\Rightarrow$ Spheromak.

- IPLOT - (obsolete).

  0.0 $\Rightarrow$ profiles plotted versus ψ.

  1.0 $\Rightarrow$ profiles plotted versus $\sqrt{\frac{V}{V_0}}$.

- IBOOT - Bootstrap current model switch.
  1.0 ⇒ collisionless Hirshman
  2.0 ⇒ collisional Harris
  3.0 ⇒ full Hirshman-Sigmar

- ICZERO - Switch to determine whether I_p or q_0 is held fixed.
  0.0 ⇒ fix q_0 to CZERO of Card 3.
  1.0 ⇒ fix I_p(MA) to CZERO of Card 3.

7.3 Card 2 - Weighted-exponentiated pressure profile

$p(\psi) = \mathrm{P0} \times [\mathrm{PET1} \times (1 - \hat{\psi}^{\mathrm{BP1}})^{\mathrm{AP1}} + \mathrm{PET2} \times (1 - \hat{\psi}^{\mathrm{BP2}})^{\mathrm{AP2}}]$
P0 = $\mu_0\times$ pressure on axis (MKS).

7.4 Card 3 - Weighted-exponentiated current profile

$\frac{<\vec{J}.\vec{B}>}{<\vec{B}.\nabla\phi>}(\psi) \sim \mathrm{CET1} \times (1 - \hat{\psi}^{\mathrm{BC1}})^{\mathrm{AC1}} + \mathrm{CET2} \times (1 - \hat{\psi}^{\mathrm{BC2}})^{\mathrm{AC2}}$

The current function normalization is with respect to the current required to meet the desired q_0 value on axis or desired I_P.

7.5 Card 4 - Specification of outer-most flux surface

- EPSU - Plasma upper elongation.
- EPSL - Plasma lower elongation.
- DUGUESS - Plasma upper triangularity.
- DLGUESS - Plasma lower triangularity.

- BGUESS - Plasma bean shape parameter (indentation).
- DZGUESS - Plasma bean shape parameter.
- ZZERO - Shift midplane by ZZERO (meters).

7.6 Card 5 - Plasma parameters

- XZERO - Plasma geometric major radius (meters).
- AGUESS - Plasma minor radius (meters).
- BTOR0 - Vacuum toroidal field at geometric major radius (Tesla).
- QSAW - value of safety factor below which sawtoothing is evoked.
  0.0 $\Rightarrow$ turn off sawtooth model
  1.0 $\Rightarrow$ if q $<$ q_{saw} sawtooth model is turned on.
- CEDGE - Value of edge current relative to peak value, 0.0 $\leq$ CEDGE $\leq$ 1.0, used in the weighted-exponentiated current profile.
- ZAVE - Average ion charge = $\frac{\sum_i Z_i n_i}{\sum_i n_i}$
  Used in resistivity and IBOOT=2 bootstrap calculations.
- ZEFF - Effective ion charge = $\frac{\sum_i Z_i^2 n_i}{\sum_i Z_i n_i}$
  Used in resistivity and IBOOT=2 bootstrap calculations.

7.7 Card 6 - Additional current drive

- ILHCD - Switch to turn on additional current source.
  0.0 $\Rightarrow$ turned off
  1.0 $\Rightarrow$ turned on

- A1CD - Total current in additional current source (MA).

- ALH - Position of maximum in current profile.

- DLH - Width of current profile.

- A1LH - Exponent in current profile function.

- A2LH - Exponent in current profile function.

$\frac{<\vec{J}.\vec{B}>}{<\vec{B}.\nabla\phi>}_{\mathrm{LHCD}}(\psi) \sim \frac{\mathrm{DLH}^2\hat{\psi}^{\mathrm{A1LH}}(1-\hat{\psi})^{\mathrm{A2LH}}}{(\hat{\psi}-\mathrm{ALH})^2+\mathrm{DLH}^2}$

7.8 Card 7 - Computation tolerances, sawtooth parameters

- FACIMP - Error tolerance for inner loop elliptic equation solver. If FACIMP is negative, | FACIMP | will be used as the error tolerance, and will be divided by 5 at each grid doubling. This can be useful in removing glitches in $q\prime$ near the axis caused by an insufficient number of iterations following grid size doubling.

- NIMAX - Maximum number of inner iterations before metric update.

- ITOOFF - Number of iterations which use fit at origin.

- JORGN - ψ surface number at which fit is made for iterations $<$ ITOOFF.

- TSAW - Sawtooth parameter for temperature profile.
  $\mathrm{T_{saw}}(\psi) = \mathrm{TSAW} \times \mathrm{T}(\psi) + (1\text{-TSAW}) \times \mathrm{T}(\psi = \psi_{\mathrm{qsaw}})$.

- ESAW - Sawtooth parameter for Taylor state current profile.
  $\frac{<\vec{J}.\vec{B}>}{<\vec{B}.\nabla\phi>}_{\mathrm{saw}}(\psi) = \mathrm{ESAW} \times \frac{<\vec{J}.\vec{B}>}{<\vec{B}.\nabla\phi>}(\psi) + (1\text{-ESAW}) \times \frac{<\vec{J}.\vec{B}>}{<\vec{B}.\nabla\phi>}(\psi_{\mathrm{qsaw}})$.

- RSAW - Relaxation parameter for sawtooth model.

7.9 Card 8 - Other parameters and switches

- TSF - Under-relaxation parameter for current update iteration.
- PHI2 - Iteration parameter used to optimize over-relaxation of inner iteration.
- SF - Iteration parameter used to optimize over-relaxation of inner iteration.

  Lowering SF can help the convergence of difficult equilibria.
- NSPEC - Number of particle species present (for IBOOT = 3 bootstrap option).
- SCALEP - Used with the TSC profiles. SCALEP > 0.0.
- DBWRT - Switch for writing scalar and profile database files.

  0.0 ⇒ Don't write dbequs, dbequp

  1.0 ⇒ Write dbequs and dbequp

  If DBWRT> 0, the first 8 letters of the top line in the inequ input file are

  used as an identifier for the equilibrium calculation.

7.10 Card 9 - Weighted-exponentiated temperature profile

$T(\psi) = \text{TZERO} \times [\text{TET1} \times (1 - \hat{\psi}^{\text{BT1}})^{\text{AT1}} + \text{TET2} \times (1 - \hat{\psi}^{\text{BT2}})^{\text{AT2}}]$

Temperature on axis = TZERO × 1 keV.

7.11 Card 10 - Weighted-exponentiated density profile

$n(\psi) = \text{DZERO} \times [\text{DET1} \times (1 - \hat{\psi}^{\text{BD1}})^{\text{AD1}} + \text{DET2} \times (1 - \hat{\psi}^{\text{BD2}})^{\text{AD2}}]$

Density on axis = DZERO $\times 10^{20} \text{m}^{-3}$.

7.12 Card 11 - Ripple loss calculation parameters

- IRIP - Switch to do alpha particle ripple calculation.
  0.0 $\Rightarrow$ turned off
  1.0 $\Rightarrow$ turned on
- NTFCOIL - Number of toroidal field coils.
- RTFCOIL - Major radius of outboard leg of TF coils (meters).
- FDT - Fraction of ions that are DT fuel ions.
- NPITCH - Number of pitch angles in velocity space examined for ripple loss.

7.13 Card 12 - Species masses in Hirshman-Sigmar bootstrap model (kilograms)

NOTE: NSPEC of card 8 must equal the number of species present (including electrons).

7.14 Card 13 - Species charges in Hirshman-Sigmar bootstrap model (Coulombs)

For charge neutrality, $\sum_j \mathrm{DZER_j} \times \mathrm{XCHG_j}$ must $= 0$.

7.15 Card 14 - Species central temperatures in Hirshman-Sigmar bootstrap model (keV)

7.16 Card 15 - Species central densities in Hirshman-Sigmar bootstrap model ($10^{20}\mathbf{m^{-3}}$)

If IBOOT=3, the parameters specified in cards 13,14 and 15 determine Z_{eff}, Z_{ave}, P0, TZERO, and DZERO.

7.17 Card 16 - Splined pressure profile function($\hat{\psi}$)

All normalizations that follow in this card are with respect to P0, the rationalized pressure on axis, specified in Card 2.

- PPNAA - Pressure profile normalized amplitude on axis.
- PPNSA - Pressure profile normalized slope (first derivative) on axis.
- PPNAC - Pressure profile normalized amplitude at PPPSNC.
- PPNSC - Pressure profile normalized slope in central region.
- PPNAE - Pressure profile normalized amplitude at edge.
- PPNSE - Pressure profile normalized slope at edge.
- PPPSNC - Pressure profile $\hat{\psi}$ at which the pressure profile central amplitude and slope are determined.

7.18 Card 17 - Splined current profile function($\hat{\psi}$)

Current profile is actually the profile of the ratio of the flux surface averaged quantities $\frac{<\vec{J}.\vec{B}>}{<\vec{B}.\nabla\phi>} = \frac{<\vec{J}.\vec{B}>}{<B_{toroidal}/R>}$. The normalizations that follow in this card are typically with respect to the on axis current required to meet the desired q value on axis, CZERO, specified in Card 3.

- CPNAA - Current profile normalized amplitude on axis.
- CPNSA - Current profile normalized slope (first derivative) on axis.
- CPNAC - Current profile normalized amplitude at CPPSNC.
- CPNSC - Current profile normalized slope in central region.
- CPNAE - Current profile normalized amplitude at edge.
- CPNSE - Current profile normalized slope at edge.
- CPPSNC - Current profile $\hat{\psi}$ at which the current profile central amplitude and slope are determined.

7.19 Card 18 - Splined temperature profile function($\hat{\psi}$)

All normalizations that follow in this card are with respect to TZERO, the temperature on axis, specified in Card 9.

- TPNAA - Temperature profile normalized amplitude on axis.
- TPNSA - Temperature profile normalized slope (first derivative) on axis.
- TPNAC - Temperature profile normalized amplitude at TPPSNC.
- TPNSC - Temperature profile normalized slope in central region.
- TPNAE - Temperature profile normalized amplitude at edge.
- TPNSE - Temperature profile normalized slope at edge.
- TPPSNC - Temperature profile $\hat{\psi}$ at which the temperature profile central amplitude and slope are determined.

7.20 Card 19 - Splined density profile function($\hat{\psi}$)

All normalizations that follow in this card are with respect to DZERO, the density on axis, specified in Card 10.

- DPNAA - Density profile normalized amplitude on axis.
- DPNSA - Density profile normalized slope (first derivative) on axis.
- DPNAC - Density profile normalized amplitude at DPPSNC.
- DPNSC - Density profile normalized slope in central region.
- DPNAE - Density profile normalized amplitude at edge.
- DPNSE - Density profile normalized slope at edge.
- DPPSNC - Density profile $\hat{\psi}$ at which the density profile central amplitude and slope are determined.

[1] DeLucia, J., Jardin, S. C., Todd, A. M. M., *J. Comput. Phys.* **37** 183 (1980)

-bb-error = =

Figure 1: Interpolation of 65 times 64 arrays X and Z onto a 65 times 5 grid using the jsoInterp2d routine.