

XNS user guide (version 3.0)

XNS: a numerical solver in the XCFC approximation for axisymmetric equilibria of polytropic Neutron Stars with differential rotation and/or magnetic field

N. Bucciantini

INAF - Osservatorio Astrofisico di Arcetri, L.go Fermi 5, Firenze, Italy.

Email: niccolo@arcetri.astro.it

L. Del Zanna

Dipartimento di Fisica e Astronomia, Università di Firenze, via G. Sansone 1, Sesto F.no (Firenze), Italy

Email: ldz@arcetri.astro.it

A.G. Pili

Dipartimento di Fisica e Astronomia, Università di Firenze, via G. Sansone 1, Sesto F.no (Firenze), Italy

Email: pili@arcetri.astro.it

1. Introduction

The XNS code solves for axisymmetric equilibria of polytropic magnetized and/or rotating Neutron Stars (NSs) using the *extended conformally flat condition* (XCFC) for the metric, in spherical coordinates. This is based on the metric module and the routines developed for the X-ECHO code for GRMHD in dynamical spacetimes (Bucciantini & Del Zanna 2011), which in turn is an upgrade of the *Eulerian conservative high-order* code (ECHO Del Zanna et al. 2007) for GRMHD in a static background metric (the so-called Cowling approximation). Like ECHO and X-ECHO, also XNS is written in the Fortran90 programming language. The reader is referred to the above cited paper for full derivation of the GRMHD equations, and for the full description of the XCFC solvers. The following guide is based on the papers Pili et al. (2014); ?, (2017), where the equations describing the approach for magnetized models is fully presented. The various quantities in this guide are referred to the one used and defined in this latter work.

If you use this software please reference the following papers:

Bucciantini N., Del Zanna L., 2011, A&A, 528, A101

Pili A.G., Bucciantini N. & Del Zanna L., 2017, MNRAS, 470, 2469

Pili A.G., Bucciantini N. & Del Zanna L., 2015, MNRAS, 447, 2821

Pili A.G., Bucciantini N. & Del Zanna L., 2014, MNRAS, 439, 3541.

2. Description of the XNS package

In the following we will list and describe the user parameters, arrays, files, subroutines, and outputs of the XNS package.

2.1. User parameters

The input parameters that the user might want to change are all set in the module `system` inside the file `SYSTEM.f90`. There are other parameters in other parts of the code that deal with specific routines (root finding, convergence etc...), but those should not need to be changed. Here is a list of the parameters of the model as they appear in the module `system`:

- **NVALUE** – the maximum number of loops employable by the Newton-Raphson scheme in the search for a equilibrium solution, having a target value for a desired quantity (central density, total mass, etc..) by the program `XNS`. Usually convergence is reached within about 10 steps, unless the NS is strongly distorted (fast rotation, and/or strong magnetic field). The default value is set to 100.
- **MAXLOOP** – the maximum number of loops employable in the search for a converged equilibrium solution by the subroutine `XNSMAIN`. Usually convergence is reached within the first 100 steps, unless the NS is strongly distorted (fast rotation, and/or strong magnetic field). The default value is set to 1000.
- **NR** – the number of radial grid points (the default setting is for a uniform grid). The radial grid is defined from $r = RMIN$ and $r = RMAX$.
- **NTH** – the number of angular grid points (the default setting is for a uniform grid). The angular grid is always defined between $\theta = 0$ and $\theta = \pi$.
- **RMIN** – the lower boundary in the radial direction. It must be always set to 0, since the metric solver requires a compact domain and has been implemented with specific boundary conditions for $RMIN = 0$.
- **RMAX** – the maximum radius of the computational domain. This can be arbitrarily chosen. However, one needs to guarantee that the NS is properly resolved over a sufficient number of grid points (50-100), so this parameter and **NR** should be chosen consistently. In particular, the condition $RMAX > 2r_{TOV}$ must hold, where r_{TOV} is the NS radius of the initial TOV guess. This is because the TOV solver is designed to converge when the ADM masses measured at **RMAX** and **RMAX/2** (hence it must be outside the NS) coincide within a given tolerance. If not the code will halt with a warning.
- **STRETCH** – a logical flag that control whether the grid is stretched or not. If `.TRUE.` the radial grid is regular up to **RREG** with **NRREG** grid points and it is stretched from **RREG** to **RMAX** with **NR-NRREG** points. The stretching factor **STRR** is determined by the code consistently with the choices for **NR**, **NRREG**, **RMAX** and **RREG**. See also Pili et al. (2015) for details.

- NRREG – number of grid points for the regular grid if STRETCH=.TRUE., otherwise it can be safely set to zero.
- RREG – maximum radius of the regular grid. It is unused if STRETCH=.FALSE..
- VERBOSE – a logical flag. Setting VERBOSE=.true. forces the code to output on screen all the INFOs related to the various steps done by each subroutine (to be used only for debugging or checks). Otherwise setting VERBOSE=.false. the output on screen will be produced only at the end. The latter is the default option.
- WRT – a logical flag. Setting WRT=.true. forces the code and each subroutine to write output files at every step or substep, otherwise setting WRT=.false. will prevent IO writing. The latter is the default option.
- WRTF – a logical flag. Setting WRTF=.true. override WRT for the final step, and allow to write all the files related to the final configuration. Setting WRTF=.false. will prevent IO writing.
- CHUP – a logical flag. Setting CHUP=.true. allows (subject to WRT, WRTF) to write the files containing the results of the metric solver and primitive solver XShiftphi.dat, Conformal.dat, Primitive.dat, Primitive_mag.dat, Shiftphi.dat, Lapse.dat, Source.dat, TOVini.dat, Rhovec.dat. Setting CHUP=.false. will prevent from writing these files. The latter is the default option, unless one wishes to perform a check of the metric or primitive solvers.
- OMG – the value of the angular velocity at the center Ω_c .
- A2VALUE – the value of A^2 . This is needed only for differentially rotating models, otherwise it should be set to 0.
- DIFFERENTIAL – a logical flag that states whether the model is differentially rotating or not. Setting it to .false. implies uniform rotation, with $\Omega = \text{OMG}$. Setting it to .true. implies differential rotation. In this case, a value of A2VALUE must be specified.
- IMAG – a logical flag that states whether the model is magnetized or not. Setting it to .false. implies the non magnetized case. Setting it to .true. implies the presence of a magnetic field. In the latter case, values of parameters for the magnetic model must be specified.
- ITOR – a logical flag that must be set true only for purely toroidal configurations.
- IPOL – a logical flag that must be set true only for purely poloidal configurations.
- ITWT – a logical flag that must be set true only for mixed Twisted Torus configurations.
- RHOINI – the central density for the starting guess. **WARNING:** this is not the central density of the converged model, it is just its value for the starting guess. By default XNS will search for a solution in the range 0.8-1.2RHOINI. If the desired solution is outside this range, XNS will output a warning, and stop.

- QUOC – the quantity of interest to which the model must converge [0 for a given central density, 1 for a given gravitational mass, 2 for a given barionic mass].
- QUCONV – the value of the quantity of interest to which we want a model to converge. For example if one wants a model with central density 1.28×10^{-3} (in geometrized units) set: QUOC=0, QUCONV=1.28E-3.
- K1 – the polytropic coefficient K of the EoS. The code uses a polytropic EoS. The value K1=100 is for the standard case used for many tests of NS stability and evolution in the literature (see also the parameter below).
- GAMMA – the adiabatic index $\gamma = 1 + 1/n$ of the polytropic EoS, where n is the polytropic index. The value GAMMA=2 ($n = 1$) is for the standard case used for many tests of NS stability and evolution in the literature (see also the parameter above).
- BCOEF – the value of K_m in the magnetic polytropic law for the case of purely toroidal field. Never used when IMAG=.false. or ITOR=.false., though it is better set to 0 in this case.
- MAGIND – the value of m in the magnetic polytropic law for the case of purely toroidal field. It must be > 1 , otherwise the magnetic energy diverges on the polar axis. It is never used when IMAG=.false. or ITOR=.false..
- KBPOL – the value of K_{pol} in the magnetic law for the case of purely poloidal field. Never used when IMAG=.false. or IPOL=.false., though it is better set to 0 in this case.
- CSI – the value of the parameter ξ (non linear current term) in the magnetic polytropic law for the case of purely poloidal field. It is never used when IMAG=.false. or IPOL=.false..
- CTP – if set to .FALSE. the code avoid to use conservative to primitive routines. This flag is effective only with IPOL=.TRUE..
- QNULL – logical flag that regulates the global net charge of a rotating star with poloidal magnetic field. If QNULL=.TRUE. the code searches for a globally uncharged star, otherwise it minimizes the electric field at the stellar pole.
- KBTT – the value of K_{pol} in the magnetic law for the case of mixed Twisted Torus configuration. Never used when IMAG=.false. or ITWT=.false., though it is better set to 0 in this case.
- ATWT – the value of a in the magnetic law for the case of mixed Twisted Torus configuration. Never used when IMAG=.false. or ITWT=.false., though it is better set to 0 in this case.
- ZETA – the value of ζ in the magnetic law for the case of mixed Twisted Torus configuration.
- CUT – the value of λ in the magnetic law for twisted magnetosphere models. It regulates the extension of the twist. If $\lambda = 1$ standard Twisted Torus models are recovered.

- **CONV** – a convergence parameter for the Newton-Raphson scheme in XNS. It is given in relative terms (beware that the code accuracy is $\sim 10^{-3}$).
- **REQMAX** – the maximum radius beyond which any NS model will be artificially truncated. This must be set \geq of the shedding mass limit. Sometimes, when working with configurations close to mass shedding, during the convergence loop the code might get unbounded solutions or fail to converge. To avoid this, setting a value for REQMAX will force the solution to be truncated. The value REQMAX= 11.6 is chosen for the BU series (Stergioulas et al. 2004), provided with the package.
- **QFACTOR** – a damping factor of the convergence loop for solving the Bernoulli Eq.n, used in HYDROEQ.f90. At the end of each sub-loop of the convergence scheme, a new set of equilibrium fluid variables is computed. Setting QFACTOR=1 implies that these will be used, while setting QFACTOR=0 means that the old variables \mathcal{V}_{old} will be used (the code will never converge in this case!). A value $0 < Q_F < 1$ implies that at the beginning of each loop a combination of new and old variables will be used, in the form

$$\mathcal{V} = Q_F \mathcal{V}_{new} + (1 - Q_F) \mathcal{V}_{old}.$$

Using a value less than 1 tends to give slower but more stable convergence. Values QFACTOR < 0.5 are to be used only for pathological cases where the convergence is very slow or when the code fails to converge (i.e. rotating models on the unstable branch of NS mass-radius curve).

- **QAPHI** – a damping factor of the convergence loop for solving the Grad-Shafranov or the Maxwell-Ampère equation. Analogous to QFACTOR but for the ϕ -component of the 4-potential.
- **EPS** – a tolerance value. It is used in several subroutines and must be a small value. This should not need to be changed.
- **MLS** – number of spherical harmonics (Legendre polynomials) for spectral decomposition in θ (numbered from 0 to MLS). This should be $< NTH$. In 1D (NTH=1) it must be set to 0.
- **NGQ** – the number of interpolation points for the Gauss quadrature, needed to compute the integrals over the polar direction of the source terms in the spherical harmonics decomposition. Used by all the 2D elliptical PDE solvers. It must be $NGQ \leq NTH$. In 1D (NTH=1) it must be set to 1.
- **MLSL** – number of spherical harmonics used to solve the Laplace equation as described in Pili et al. (2017). This parameter is relevant only in the cases of rotating and poloidal magnetized star. It is set to 10 by default.
- **TOLCONV** – the convergence tolerance for the iterative solution of the PDEs for the conformal factor ψ and the lapse α .

2.2. Arrays

The module `system` inside the file `SYSTEMXNS.f90` also contains the definitions of some arrays that are used within the code and shared by many subroutines. We briefly describe some of them here so that the user can have some idea of what represents what. Other arrays that are specific only to certain subroutines are defined locally and are not discussed here. Notice that some arrays related to the poloidal components of the velocity, shift vector, or auxiliary vectors are always zero but still defined in XNS, though related routines are never called. This is because XNS shares the same metric solver as the full X-ECHO code.

- `R, DR` – 1D arrays that store the location of the radial points, and the radial increments.
- `TH, DTH, XX` – 1D arrays that store the location of the angular points, the angular increments, and the cosine of the angle.
- `PSI, PSL, PSS, PSSR, PSST` – 2D arrays of metric terms, respectively ψ , $\alpha\psi$, X^ϕ , X^r , X^θ . We have either $X^i \equiv W^i$, or $X^i \equiv \beta^i$, depending on the step of the metric solver.
- `RHOSRC, ESRC, PSRC, VPHI, VR, VTH, BPHI, SSS` – 2D arrays, respectively ρ , $\rho h \equiv \rho(1 + \varepsilon) + p$, p , v^ϕ , v^r , v^θ , B^ϕ , S , needed for the source terms.
- `USRC, DSRC, S3SRC, S1SRC, S2SRC` – 2D arrays containing the \mathcal{U} conservative variables needed for the source terms, respectively \hat{E} , \hat{D} , \hat{S}_ϕ , \hat{S}_r , \hat{S}_θ , all multiplied by $f^{1/2} = r^2 \sin \theta$.
- `ECSRC, ELSRC, ES1RC, ES2RC, ES3RC` – 2D arrays containing the source terms (the right hand side of the equations) associated with the presence of matter in the elliptic PDEs. Respectively, the source for the equations for ψ , $\alpha\psi$, X^r , X^θ , X^ϕ , where $X^i \equiv W^i$ or $X^i \equiv \beta^i$.
- `CURVC, CURVR, CURVT, CURVP` – 2D arrays containing the source terms associated with the curvature of the metric, respectively for the two scalar Poisson equations (for ψ and $\alpha\psi$) and for the three components of the second vector Poisson equation (that for β^i).
- `MU, NU` – 1D arrays containing the metric terms of the radial TOV solution. The metric employed is that for isotropic coordinates, namely $ds^2 = -e^\nu dt^2 + e^\mu(dr^2 + r^2 d\theta^2 + r^2 \sin^2 \theta d\phi^2)$.
- `RHOTV, PRTV, ETV` – 1D arrays containing the fluid variables of the radial TOV solution, respectively ρ , p , $\rho\varepsilon$.
- `RHONew, PNew, ENew, V3New, B3New, E3New` – 2D arrays, respectively ρ , p , $\rho\varepsilon$, v^ϕ , B^ϕ , E_ϕ computed for an equilibrium configuration on the metric at the end of each step of the convergence loop.
- `BPOLR, BPOLT, EPOLR, EPOLT, APHI, ATIM` – 2D arrays, respectively, B^r , B^θ , E_r , E_θ , $\tilde{A}^\phi \equiv \Phi$, $A^t \equiv \Psi$ for the magnetic configuration with poloidal field components.

2.3. Files and Outputs

Here is a list of all the files and subroutines included in the XNS package, together with a brief description of what they do and how they operate. We start with the `fortran90` files of the code (with extension `.f90`), then we describe the output files produced by a run (with extension `.dat`), and we conclude with the IDL (Interactive Data Language) files needed for visualization (with extension `.pro`). The code must run in double precision for convergence. The `Makefile` is provided for both the `gfortran` (GNU) and `Ifort` (Intel) compilers. Precompiler options are: `serial` (standard XNS run, with no overall convergence on a specific quantities, it generates the executable `XNS-s`) and `nwtrps` (load the version of XNS with Newton-Raphson scheme, generating the executable `XNS-nr`).

- `XNS.f90` – main program. Makes some consistency checks, and invokes `XNSMAIN`. Depending on the pre-compiling option it simply call `XNSMAIN` (if `make serial` is used to compile the code), or it performs a Newton-Raphson search for an equilibrium model with a given value for a desired quantity of interest, i.e. a certain value of the central density or gravitational mass (if `make nwtrps` is used).
- `XNSMAIN.f90`
 - subroutine `xnsmain` – the main kernel of the code: it defines the grid (uniform in this version), builds a 2D initial guess based on the 1D TOV output of `TOVINI.f90`, performs the convergence loop calling all the various metric solvers and procedures in the appropriate order. When the loop is over, it writes all the outputs.
 - subroutine `conformal` – solves for the scalar Poisson-like equation for ψ .
 - subroutine `lapse` – solves for the scalar Poisson-like equation for $\alpha\psi$.
 - subroutine `shiftphi` – solves the ϕ component of the two vector Poisson equations for W^i and β^i , given the corresponding source terms.
 - subroutine `curv1` – computes the curvature source term in the routines for ψ and $\alpha\psi$.
 - subroutine `curv2` – computes the curvature source term in the routine for β^ϕ .
 - subroutine `legzo` – computes the zeros of Legendre polynomials and the corresponding weights for Gaussian quadrature integration.
 - subroutine `lpn` – computes the Legendre polynomials and their derivatives.
 - subroutine `dgtsv` – solves the linear system $AX = B$, where A is a tridiagonal matrix, by Gaussian elimination with partial pivoting (taken from the LAPACK routines).
 - subroutine `polint` – a polynomial 2nd order interpolation routine (modified from the Numerical Recipes).
- `SYSTEM.f90`
 - module `system` – contains various parameters of the run, to be specified by the user, and definitions of common arrays (see the previous sub-sections).

- subroutine `eostov` – EoS for the 1D TOV solution (here polytropic), it provides the density and the thermal energy as functions of the pressure.
 - subroutine `funcd` – used by the root-finding subroutine to derive the central pressure given the central density.
 - function `rtsafeg` – the root-finding subroutine, based on bisection and Newton’s methods (modified from the Numerical Recipes), to derive the central pressure.
- `HYDROEQ.f90`
 - subroutine `hydroeq` – given the CFC metric and a value of ρ_c it computes the equilibrium configuration for the corresponding Bernoulli integral (works for polytropic EoS). It finally calls `hydrovar_<x>` depending on the physical parameter set in `SYSTEMXNS.f90` to compute local equilibrium quantities.
 - subroutine `hydrovar`, `hydrovar_tor`, `hydrovar_pol` – they compute local equilibrium quantities such as ρ, p, v^ϕ, B^i and E_i depending on the specific choice for the magnetization (respectively unmagnetized case, purely toroidal magnetic field and poloidal magnetic field).
 - subroutine `covterm` – computes the local terms of the metric tensor
 - subroutine `cons_to_prim` – computes the inversion from conserved to primitive variables.
 - subroutine `cons_to_prim_pol` – computes the inversion from conserved to primitive variables for the specific case of poloidal field.
 - subroutine `omegavalue` – derives the function $\Omega = \Omega(r, \theta)$ for the differential rotation.
 - subroutine `quantities` – computes several quantities (e.g. mass, energy, angular momentum) at the end of the convergence loop, according to standard definitions.
 - subroutine `vecpotphi` – solve the Grad-Shafranov Equation for the ϕ -component of the vector potential
 - subroutine `sourcepot` – compute source terms (currents and metric) for the Grad-Shafranov Equation or Maxwell equations depending if the rotational rate `OMG` is set to zero or not.
 - subroutine `vecpotphi` – called by the subroutine `hydrovar_pol` when `OMG.EQ.0`, it solves the Grad-Shafranov Equation.
 - subroutine `mxwlsol` – called by subroutine `hydrovar_pol` when `OMG.NE.0`, it solves iteratively the Maxwell-Ampère and the Maxwell-Gauss equation. It finally corrects the solution for the electric potential Φ in order to guarantee that the MHD condition $\Phi = -\Omega\Psi + C$ is valid inside the star. Indeed, as explained in Pili et al. (2017), the solution for Φ obtained by solving the non-homogeneous Maxwell equations, does not satisfy the perfect conducting relation inside the star, but differs from the MHD solution $\Phi_{\text{MHD}} = -\Omega\Psi + C$ by an harmonic function Φ_a so that $\Phi = \Phi_{\text{MHD}} + \Phi_a$ with $\Delta\Phi_a = 0$. The harmonic function is obtained evoking the `laplace` subroutine.

- subroutine `laplace` – solves the equations $\Phi_a|_{\mathcal{S}_{\text{NS}}} = \sum_{\ell} Y(\theta) (a_{\ell} r^{\ell})|_{\mathcal{S}_{\text{NS}}}$ (inside the star) and $\Phi_a|_{\mathcal{S}_{\text{NS}}} = \sum_{\ell} Y(\theta) (b_{\ell} r^{-(\ell+1)})|_{\mathcal{S}_{\text{NS}}}$ (outside the star), where \mathcal{S}_{NS} is stellar surface and $\Phi_a|_{\mathcal{S}_{\text{NS}}} = (\Phi + \Omega\Psi + C)|_{\mathcal{S}_{\text{NS}}}$. Each system of equations is solved with a LU decomposition and a subsequent backward substitution adopting the routines provided in the Numerical Recipes (`ludcmp` and `lubksb`). Notice that, in order to avoid spurious effects, the surface terms are evaluated on top of the super-ellipsoid that best fit the numerical surface.
- subroutine `solveaphi` and subroutine `solveatim` – solve respectively for the Maxwell-Ampère and Maxwell-Gauss equations.
- `TOVINI.f90`
 - subroutine `tovini` – solves the 1D TOV equations in isotropic coordinates to provide the initial guess. It uses a shooting method to achieve convergence.
 - subroutine `rk4` – the 4th order RK integrator (modified from the Numerical Recipes).
 - subroutine `derivs` – provides the derivatives needed to integrate the TOV equations via the RK4 method.
 - subroutine `init` – a Taylor expansion of the TOV equations at small initial radii (they are singular for $r \rightarrow 0$).

Outputs

- `Grid.dat` – contains the mesh points.
- `TOVini.dat` – contains the 1D TOV solution $(r, \mu, \rho, \nu, p, \rho\epsilon)$
- `Source.dat` – contains 2D source term for the metric solver $(\rho, p, \rho\epsilon)$
- `XShiftphi.dat` – contains the W^{ϕ} component and the related source term of its vector Poisson equation.
- `Conformal.dat` – contains ψ and the two (matter and curvature) source terms of its scalar Poisson equation.
- `Primitive.dat` – contains the primitive variables $(\rho, p, \rho\epsilon, \nu^{\phi}, B^{\phi})$ recovered self-consistently from the metric and the conserved variables.
- `Primitive_mag.dat` – contains the magnetic primitive variables $(B^{\phi}, B^r, B^{\theta})$ recovered self-consistently from the metric and the conserved variables.
- `Lapse.dat` – contains α and the two (matter and curvature) source terms of the related scalar Poisson equation.
- `Shiftphi.dat` – β^{ϕ} vector and the related source term of the vector Poisson equation

- `Rhovec.dat` – central density at each step of the `XNSMAIN` subroutine
- `Hydroeq.dat` – contains the new equilibrium configuration $(\rho, p, \psi, v^\phi, \alpha, \beta^\phi)$
- `Hydroeq_mag.dat` – contains the new equilibrium configuration for magnetic field $(B^\phi, B^r, B^\theta, \tilde{A}^\phi, E_\phi, E_r, E_\theta, A^t, J^\phi, J^r, J^\theta)$.
- `Mxwll_test.dat` – contains data related to the source term of both Maxwell-Ampère and Maxwell-Gauss equation $(\rho_c, J^\phi, \Phi_{\text{int}}, \Phi_{\text{ext}}, \Phi_a, \omega, \Gamma)$
- `Apconv.dat` – maximum value of Ψ at each step of the `XNSMAIN` subroutine
- `Atconv.dat` – maximum value of Φ at each step of the `XNSMAIN` subroutine
- `LogFile.dat` – summary of the run (input and output quantities).

IDL visualization routines

- `xnsdata.pro` – visualizes properties of the equilibrium solution that is computed by the code.
- `imdisp.pro` – subroutine for visualization display.

3. Examples

Here we present a few examples to show how to work with the code and the related performances. All cases have been run on a simple laptop.

3.1. A 2D uniformly rotating case

Let us consider the case of a uniform rotator. We are interested in deriving an equilibrium model corresponding to a central density $\rho_c = 1.2769\text{e-}3$ and a rotation rate $\Omega_c = 0.002633$. This is the Model BU8 of Stergioulas et al. (2004). The solution will then be compared with the one derived by the RNS code (Stergioulas & Friedman 1995; Nozawa et al. 1998), which can be considered as a fiducial solution.

We chose the following setup:

```
NR = 250, NTH = 100, NVALUE = 100, MAXLOOP = 1000,
RMIN = 0., RMAX = 20., STRETCH=.FALSE.,
QUOC = 0, QUCONV = 1.277E-3, RHOINI = 1.28E-3,
K1 = 100., GAMMA = 2.
OMG = 0.02633, A2VALUE = 0.0, DIFFERENTIAL = .false.
IMAG = .false., ITOR = .false., IPOL = .false., ITWT = .false.
BCOEF = 0., MAGIND = 1., KBPOL = 0., CSI = 0., KBTT = 0., ATWT = 0.,
```

NPOL = 0., CONV = 1.E-4, REQMAX = 11.6,

QFACTOR=0.7 (using 1 fails to converge), MLS = 20, NGQ = 50, TOLCONV = 1.d-10

Notice that, since we are searching for a specif central density (QUOC = 0), the code must be compiled using make nwtrps in order to load the global iteration scheme.

We try as initial guess the value: RHOINI = 1.28e-3, quite close to the desired value. Given the oscillatory nature of the XNSMAIN convergence algorithm, it takes about 130 steps for each call of substep done by XNSMAIN to get a converged results with accuracy 10^{-7} . For this test the code must be compiled using make nwtrps. Then several quantities are provided on the new configuration, and the CPU time it took :

Stellar Quantities -

```
Gravit. Mass =      1.6913789416057985
Rest    Mass =      1.8256780716420724
Proper Mass =      1.9297345392332990
Rotat.   Energy =      2.3933124381473516E-002
Angul.  Moment. =      1.8179357676774555
Magnet. Energy =      0.000000000000000000
MEnergy Ratio =      0.000000000000000000
KEnergy Ratio =      9.1247249207515174E-002
Equatorial Radius =      11.3200000000000000  16.715814903359849  KM
Radius Ratio =      0.59010600706713778
Circ Radius   =      13.115464777544160      19.367109681357231  KM
Def.rate      =      0.24311855795287238
```

These can be compared to the values provided by the authors of RNS for the mass (1.692), and the kinetic energy ratio (0.18) [see Figs. 1-2 for a detailed comparison]. The differences are again due both to the XCFC approximation, discretization errors, ad the presence of an atmosphere. Note that despite the difference in kinetic energy the velocity and density profiles are indistinguishable.

3.2. A 2D purely toroidal case

Let us now consider a more interesting case of a static configuration with purely toroidal field. We are interested in deriving an equilibrium model corresponding to the one reported in Table 2 of Pili et al. (2014) with $m = 1$ and central density $16.85 \times 10^{14} \text{ g cm}^{-3}$. This is a maximum mass model. For this reason convergence is searched for the central density, and not for the mass. Indeed XNS looks for zeros and cannot look for maxima/minima.

We chose the following setup (to do this test the code must be compiled using make nwtrps):

```
NR = 250, NTH = 100, NVALUE = 100, MAXLOOP = 1000,
RMIN = 0., RMAX = 30., STRETCH=.FALSE.
```

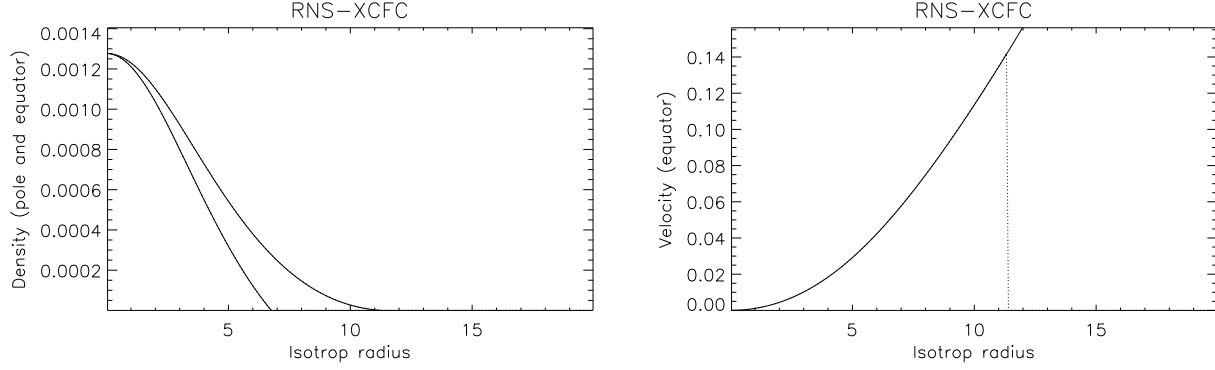


Fig. 1.— Left: comparison of the equatorial and polar density between XNS (dashed line) and RNS (solid line). The maximum deviation is $< 10^{-7}$ in absolute value, and is mostly due to the truncation at the stellar surface. Right: comparison of the rotational velocity at the equator between XNS (dashed line) and RNS (solid line) [note that RNS extends the velocity profile also to the atmosphere].

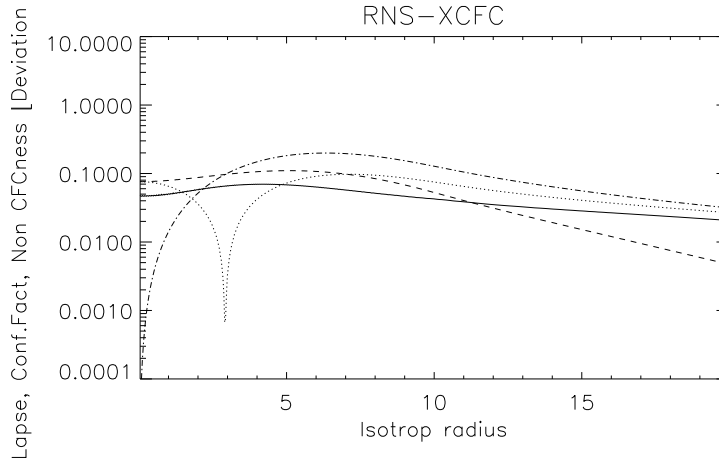


Fig. 2.— Errors of the equatorial lapse and conformal factor between RNS and XNS. Solid line is the lapse; dotted and dashed lines are the conformal factors, note that RNS uses quasi-isotropic coordinates and the coefficient ψ^4 multiplying $(dr^2 + r^2 d\theta^2)$ differs from $R/(r^2 \sin(\theta)^2)$ multiplying $d\phi^2$, so we plot the difference with both; dot-dashed line is the difference between ψ^4 and $R/(r^2 \sin(\theta)^2)$, given by RSN which can be considered as a measure of the non conformally-flatness of the solution (the XCFC intrinsic approximation error). Note that XNS errors are of order of the XCFC approximation errors.

QUOC = 0, QUCONV = 2.7164E-3, RHOINI = 2.513E-3,
K1 = 110., GAMMA = 2.
OMG = 0.0, A2VALUE = 0.0, DIFFERENTIAL = .false.
IMAG = .true., ITOR = .true., IPOL = .false., ITWT = .false.
BCOEF = 5.806, MAGIND = 1., KBPOL = 0., CSI = 0., KBTT = 0., ATWT = 0.,
NPOL = 0., CONVF = 1.E-4, REQMAX = 28.0,
QFACTOR=0.5 (using 1 fails to converge), MLS = 20, NGQ = 50, TOLCONV = 1.d-10

We try as initial guess the value: RHOINI = 2.513E-3, quite close to the best guess, in fact convergence is achieved within a few steps. Then several quantities are provided on the new configuration, and the CPU time it took:

Stellar Quantities -

Gravit. Mass =	1.9461801125091531	
Rest Mass =	2.0411290086614193	
Proper Mass =	2.2022317039068002	
Rotat. Energy =	0.0000000000000000	
Angul. Moment. =	0.0000000000000000	
Magnet. Energy =	0.10810760965626250	
MEnergy Ratio =	0.29686908732057110	
KEnergy Ratio =	0.0000000000000000	
Equatorial Radius =	12.300000000000001	18.162943755417505 KM
Radius Ratio =	1.1365853658536584	
Circ Radius =	14.240642791734739	21.028617403843072 KM
Def.rate =	-1.1520720810951663	
Mag. Flux =	1.4785869645379783	2.6930780474785728E+022 Wb,
Bmax =	1.5095656480911322E-002	1.2607024455370698E+018 G

Note that the differences in the various radii and related quantities between this run and the one reported in Table 2 by Pili et al. (2014) are due to truncation errors. Indeed they are equal to $R_{MAX}/NR = 0.12$ and in relative terms $\sim 5 \times 10^{-3}$. In Fig. 3 the density and magnetic field distribution are shown using the `xnsdata.pro` IDL program.

3.3. A 2D purely poloidal case

Let us now consider a static configuration with purely poloidal field. We are interested in deriving an equilibrium model corresponding to the one reported in Table 4 of Pili et al. (2014) and central density $16.76 \times 10^{14} \text{ g cm}^{-3}$. Again, this is a maximum mass model. For the same reason as before, convergence is searched for the central density, and not for the mass. XNS looks for zeros and cannot look for max-

ima/minima. We chose the following setup (use `make nwtrps` to compile the code):

```
NR = 250, NTH = 100, NVALUE = 100, MAXLOOP = 1000,  
RMIN = 0., RMAX = 25., STRETCH=.FALSE.,  
QUOC = 0, QUCONV = 2.7025E-3, RHOINI = 2.75E-3,  
K1 = 110., GAMMA = 2.  
OMG = 0.0, A2VALUE = 0.0, DIFFERENTIAL = .false.  
IMAG = .true., ITOR = .false., IPOL = .true., ITWT = .false.  
BCOEF = 0.0, MAGIND = 0., KBPOL = 0.437, CSI = 0., KBTT = 0., ATWT = 0.,  
NPOL = 0., CONVF = 1.E-4, REQMAX = 18.0,  
QFACTOR=0.5 (using 1 fails to converge), MLS = 20, NGQ = 50, TOLCONV = 1.d-10
```

We try as initial guess the value: $RHOINI = 2.75E-3$, quite close to the best guess, in fact convergence is achieved within a few steps. Then several quantities are provided on the new configuration, and the CPU time it took :

Stellar Quantities -

Gravit. Mass =	1.7598327977363997	
Rest Mass =	1.9176439597453208	
Proper Mass =	2.1837344396429783	
Rotat. Energy =	0.0000000000000000	
Angul. Moment. =	0.0000000000000000	
Magnet. Energy =	2.9955923403271075E-002	
MEnergy Ratio =	6.6002917419301116E-002	
KEnergy Ratio =	0.0000000000000000	
Equatorial Radius =	6.2500000000000000	9.2291380871023900 KM
Radius Ratio =	0.8559999999999998	
Circ Radius =	8.1621102465780666	12.052678807651571 KM
Def.rate =	0.14725262994099331	
Mag. Dipole =	0.39312249837664015	
Bmax =	1.3937856539534466E-002	1.1640096505345989E+018 G

Note that, as before, the differences in the various radii and related quantities between this run and the one reported in Table 4 by Pili et al. (2014) are due to differences in the settings (higher resolution and stronger convergence was imposed in those model), and different truncation in the accuracy we chose for the density and other quantities. Such differences however are a few 10^{-3} on almost all quantities and only $\sim 1\%$ on the magnetic terms. In Fig. 4 the density and magnetic field distribution are shown using the `xnsdata.pro` IDL program.

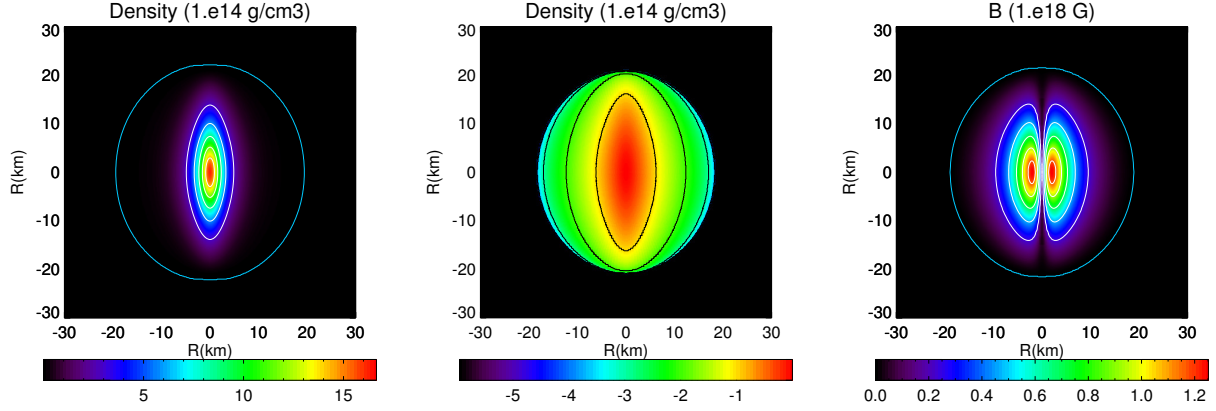


Fig. 3.— Left panel: baryonic density in linear scale. Central panel: baryonic density in Log10 scale. Right panel: magnetic field strength in linear scale.

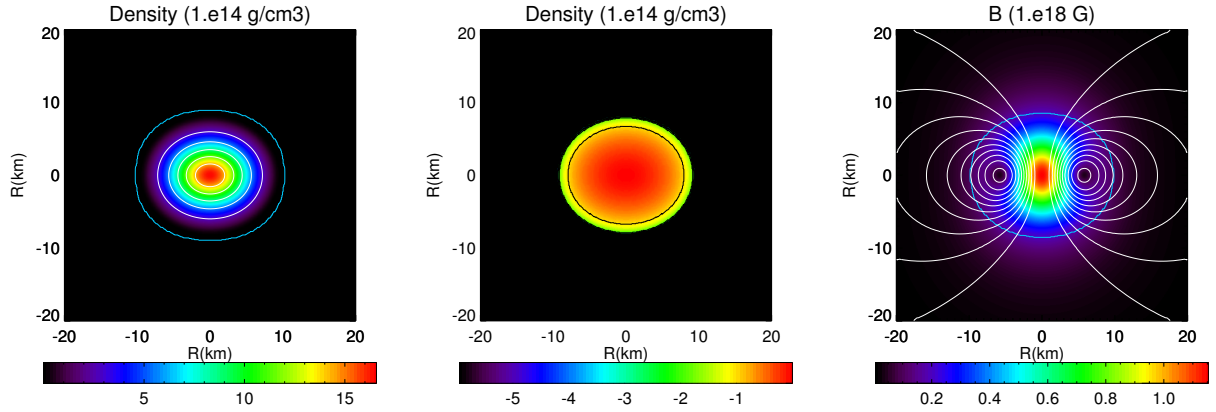


Fig. 4.— Left panel: baryonic density in linear scale. Central panel: baryonic density in Log10 scale. Right panel: magnetic field strength in linear scale.

3.4. A rotating star with a purely poloidal magnetic field

We now discuss a model that includes both rotation and poloidal magnetic field. We stress that this version of XNS is not designed to work with rotating mixed field configuration but only purely toroidal or purely poloidal field can be chosen together with rotation. Here we derive the equilibrium configuration presented in Table 4. of Pili et al. (2017) with maximum magnetic field strength 5.72×10^{17} G.

In order to reduce the computational time of the test, we suggest to drop the Newton-Raphson global convergence compiling the code with `make serial`. We directly provide the right setup to obtain the required solution:

```
NR= 600, NTH = 300, NVALUE =100, MAXLOOP = 1000,
RMIN = 0., RMAX = 30., REQMAX = 15.0, STRETCH=.FALSE.,
RHOINI = 7.4431E-004, K1 = 110., GAMMA = 2.,
OMG = 5.e-3, DIFFERENTIAL = .FALSE., A2VALUE = 0.0,
IMAG = .TRUE., ITOR = .FALSE., IPOL = .TRUE., ITWT = .FALSE.
KBPOL = 0.3973242, CSI = 0., QNULL=.TRUE., CTP=.FALSE.,
MLS = 40, NGQ = 80, MLSL= 20 , QFACTOR=0.5, QAPHI=1.,
EPS=1.E-7, TOLCONV = 1.D-10.
```

We obtain the following quantities:

```
Gravit. Mass = 1.5506213174258330
Rest Mass = 1.6424667614658279
Proper Mass = 1.7182455199175202
Rotat. Energy = 9.6505051682437427E-004
Angul. Moment. = 0.39211162557271256
Magnet. Energy = 2.7902253467743068E-002
MEnergy Ratio = 0.14200233876834248
KEnergy Ratio = 4.9114108499188364E-003
Equatorial Radius = 10.324999999999999 15.246536119893147 KM
Radius Ratio = 0.63680387409200978
Circ Radius = 12.015042216707855 17.742157398457856 KM
OMGcen = 5.0000000000000001E-003
OMGeq = 5.0000000000000001E-003
Mag. Dipole = 0.91968584612156790
Bmax = 6.8535047027379770E-003 5.7236531251007693E+017 G
B@pole = 2.3606635430019137E-003 1.9714904784141261E+017G
Def.rate = 0.31790200027471760
```

In the left and central panels of Fig. 5 we show the density, the electric and the magnetic field distribution as obtained with the `xnsdata.pro` IDL program.

3.5. Twisted Magnetosphere

As a final test we present a NS endowed with a twisted magnetosphere. This allow us to discuss how to initialize an irregular grid as the one presented in Pili et al. 2015. In particular we want to obtain a NS with a twisted magnetosphere extending up to four stellar radii adopting the current distributions prescribed, again, in Pili et al. 2015 (see in particular equations 10 and 11 of the cited paper). This prescription on the currents free function has been chosen as a default for the present version of the code in the case of mixed field configuration. However, other possible choices (as those presented in Pili et al. 2014, Bucciantini et al. 2015 and references therein) can be implemented by modifying the source term of the Grad-Shafranov equation (in particular TERM1 and TERM4 with ITWT=.TRUE. in the VECPOTPHI subroutine) and the expressions for the toroidal magnetic field and the current density (B3NEW, JRR, JTH and JPHI respectively at lines 357, 376, 380 and 382 of HYDROEQ.f90 in the HYDROVAR_POL subroutine) accordingly.

The initialization of the code is the following:

```
NR= 600, NTH = 300, NVALUE =100, MAXLOOP = 1000,
RMIN = 0., RMAX = 80., REQMAX = 15.0,
STRETCH=.TRUE., NRREG=300, RREG=20.,
RHOINI = 1.38E-3, K1 = 110., GAMMA = 2.,
OMG = 0.0, DIFFERENTIAL = .FALSE., A2VALUE = 0.0,
IMAG = .TRUE., ITOR = .FALSE., IPOL = .FALSE., ITWT = .TRUE.
KBTT=3.4E-004, ATWT=1.4E-003, ZETA=0.0, CUT=4.,
MLS = 40, NGQ = 80, QFACTOR=1., QAPHI=1.,
EPS=1.E-7, TOLCONV = 1.D-10.
```

With the flag STRETCH set to .TRUE. the XNS code builds a grid that is regularly spaced up to a radius RREG with NRREG radial grid points, and it is uniformly stretched from RREG to RMAX with (NR-NRREG) grid points and with a geometrical stretching ratio that is automatically computed by the code (listed in output in the Logfile.dat as STR).

With this initialization we obtain the following quantities:

```
Gravit. Mass = 1.5500667797561587,
Rest Mass = 1.6777495847029595,
Proper Mass = 1.7964441248872729,
Magnet. Energy = 2.3564563496900486E-008,
MEnergy Ratio = 9.5644188513138740E-008,
Equatorial Radius = 8.033333333333332 11.862518821288939 KM,
Radius Ratio = 1.0000000000000000,
Circ Radius = 9.6588808535873785 14.262903226340372 KM,
Mag. Flux = 9.7338287561287665E-004 1.7729062388452243E+019 Wb,
Mag. Dipole = 1.2222307023858927E-003,
Bmax = 7.7724764125643341E-006 649112546618277.12 G,
B@pole = 1.6105575803360884E-006 134504510140084.72 G,
Poloidal En = 2.1267399350246760E-008,
```

Toroidal En = 2.2971641466541446E-009,
Tor./Total = 9.7483840384155515E-002,

The profile of both the toroidal and poloidal component of the magnetic field for this model is shown in the right panel of Fig. 5.

REFERENCES

- Bucciantini N., Del Zanna L., 2011, A&A, 528, A101
- Bucciantini, N., Pili, A. G., Del Zanna, L., 2015, MNRAS, 447, 3278
- Del Zanna L., Zanotti O., Bucciantini N., Londrillo P., 2007, A&A, 473, 11
- Pili A. G., Bucciantini N., Del Zanna L., 2017, MNRAS, 470, 2469
- Pili A. G., Bucciantini N., Del Zanna L., 2015, MNRAS, 447, 2821
- Pili A. G., Bucciantini N., Del Zanna L., 2014, MNRAS, 439, 3541
- Nozawa, T., Stergioulas, N., Gourgoulhon, E., & Eriguchi, Y. 1998, A&AS, 132, 431
- Stergioulas, N., Apostolatos, T. A., & Font, J. A. 2004, MNRAS, 352, 1089
- Stergioulas, N. & Friedman, J. L. 1995, ApJ, 444, 306

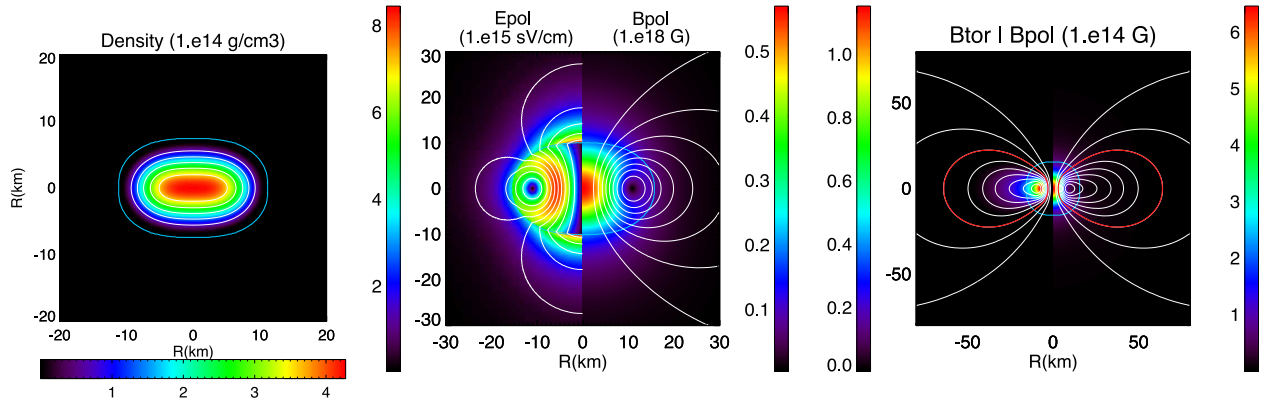


Fig. 5.— Left and center panels: baryonic density, strength of the poloidal electric field (left-half of central panel) and strength of the poloidal magnetic field (right-half of central panel) for the configuration discussed in section 3.4. Right panel: toroidal and poloidal magnetic field strength for the configuration discussed in Section 3.5. Here the red lines represent the boundary of the twisted magnetosphere, where the toroidal component of the magnetic field does not vanish.