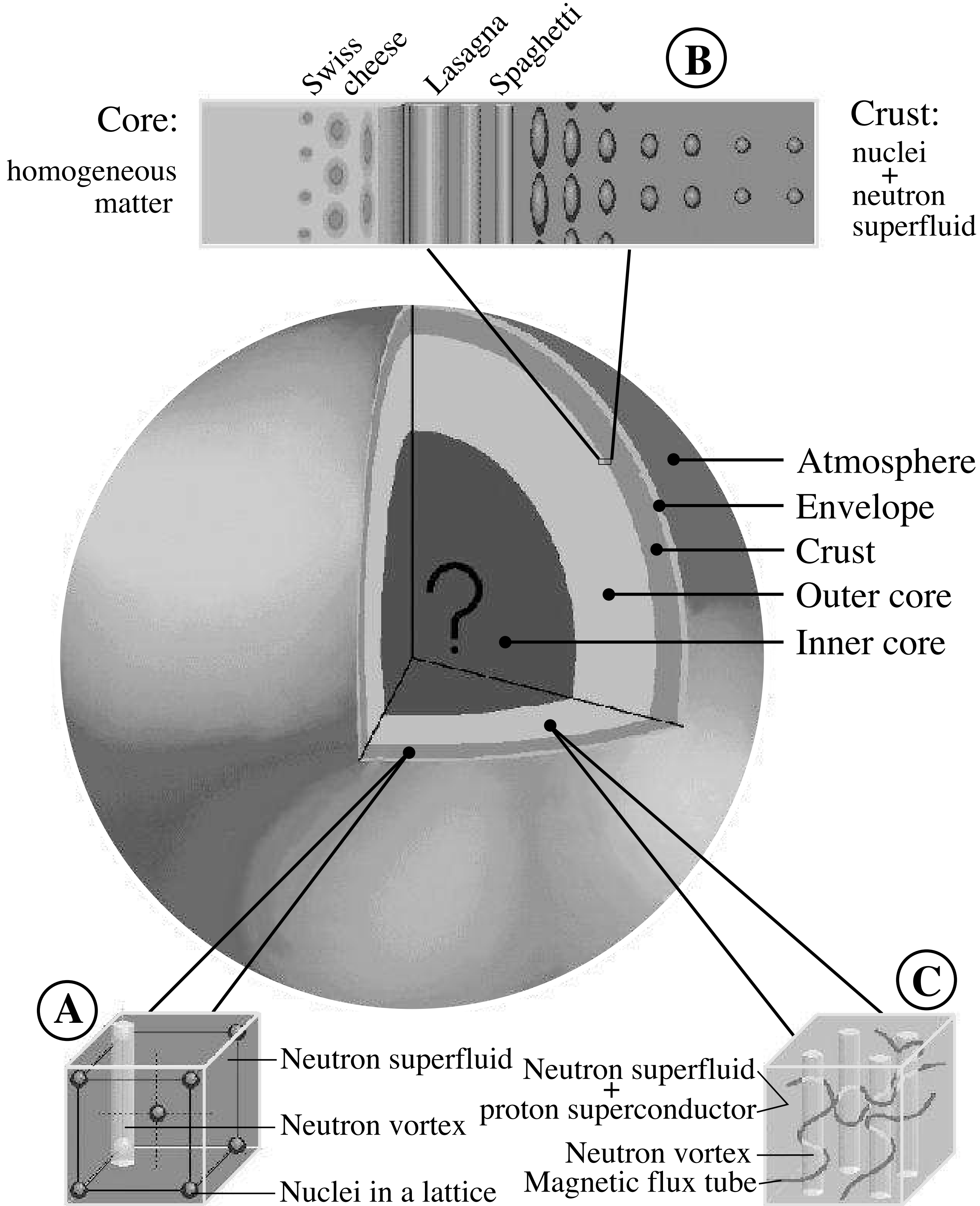
Cooling of Neutron Stars

Dany Page
Instituto de Astronomía
Universidad Nacional Autónoma de México

NSCool web site

http://www.astroscu.unam.mx/neutrones/NSCool/

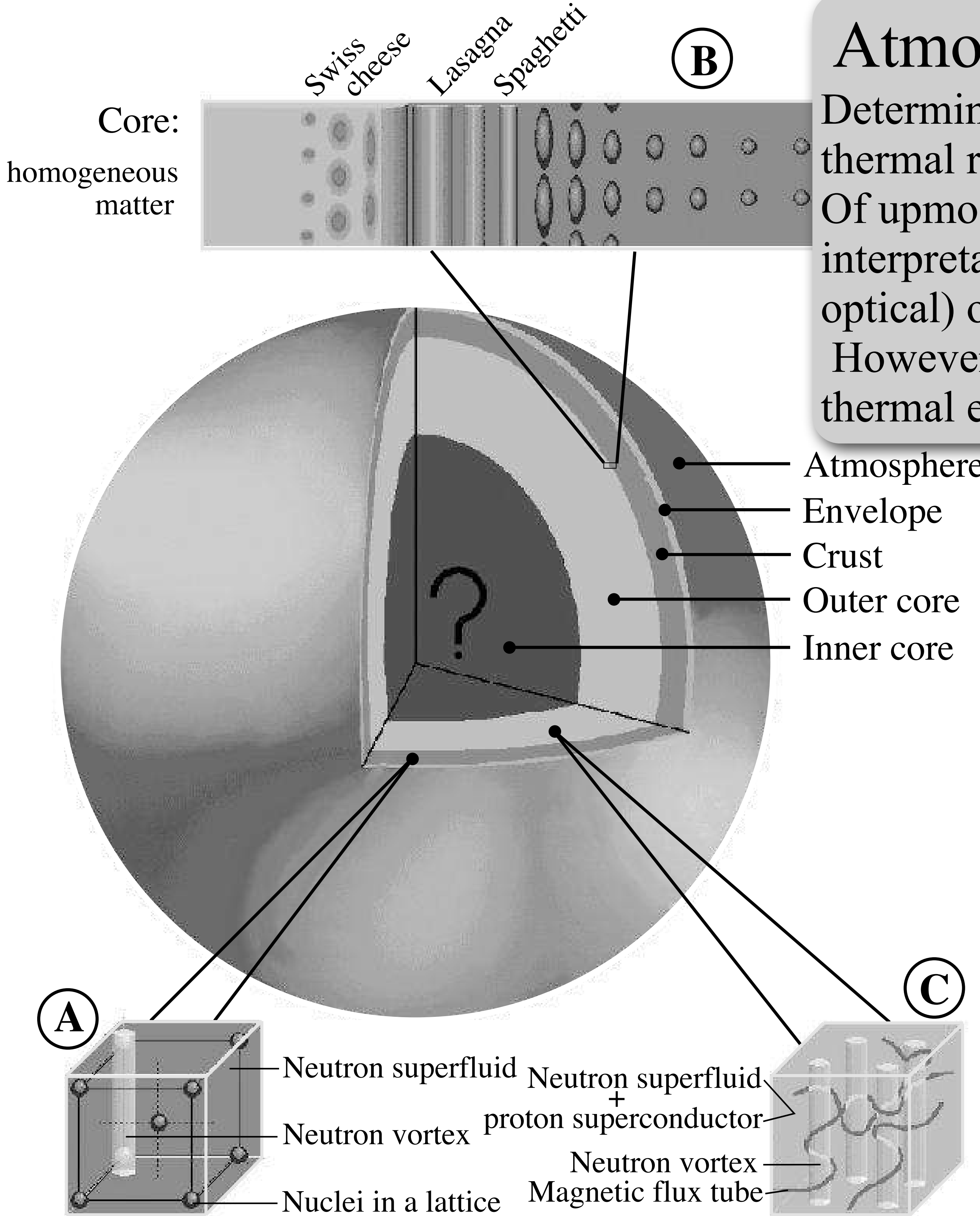


Dense Matter in Compact Stars: Theoretical Developments and Observational Constraints, Page D. & Reddy S., 2006, Annu. Rev. Nucl. Part. Sci. 56, 327

Atmosphere (10 cm):

Determines the shape of the thermal radiation (the spectrum). Of upmost importance for interpretation of X-ray (and optical) observation.

However it as NO effect on the thermal evolution of the star.



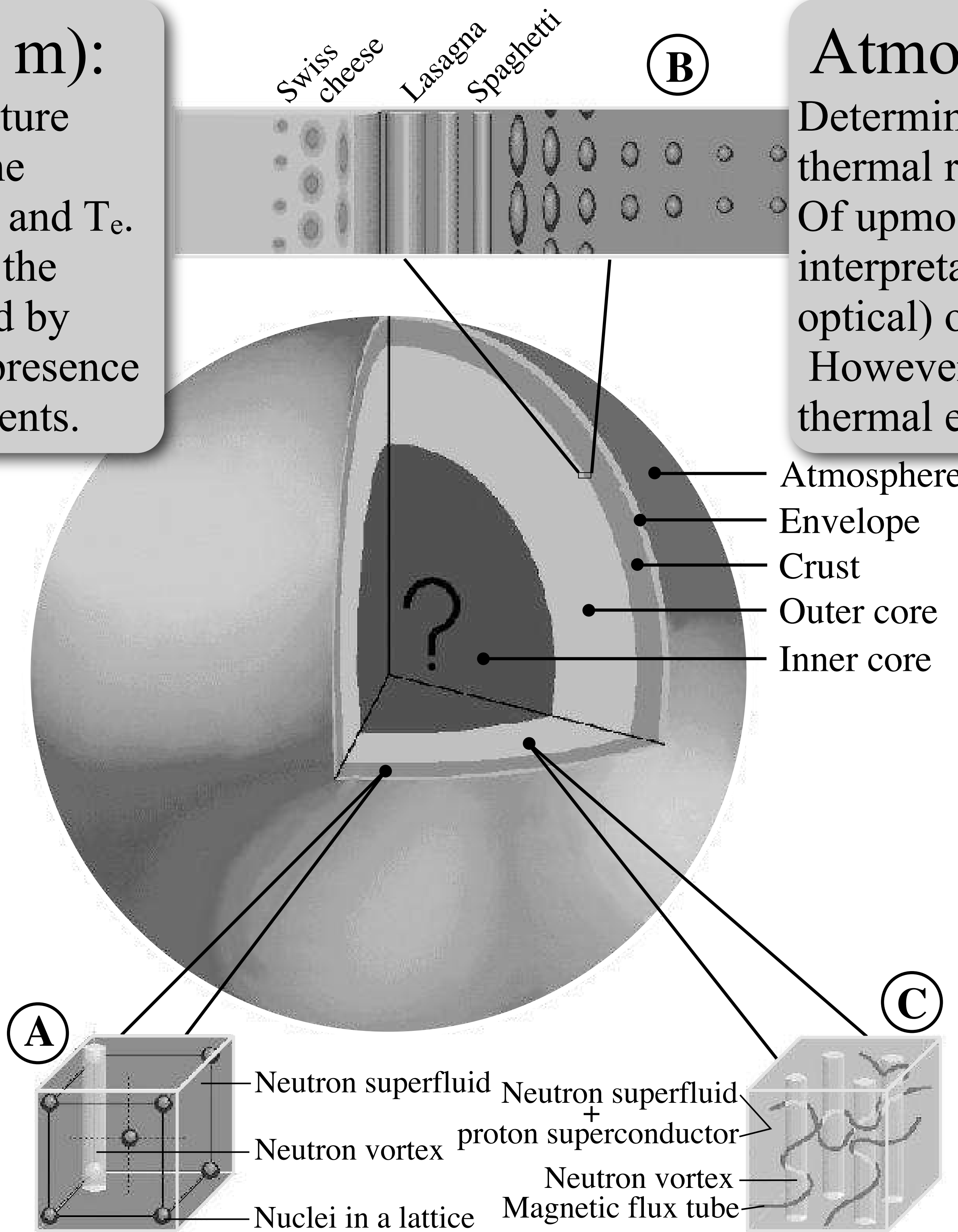
Dense Matter in Compact Stars: Theoretical Developments and Observational Constraints, Page D. & Reddy S., 2006, Annu. Rev. Nucl. Part. Sci. 56, 327

Envelope (100 m):

Contains a huge temperature gradient: it determines the relationship between T_{int} and T_e. Extremely important for the cooling, strongly affected by magnetic fields and the presence of “polluting” light elements.

Atmosphere (10 cm):

Determines the shape of the thermal radiation (the spectrum). Of upmost importance for interpretation of X-ray (and optical) observation.
However it as NO effect on the thermal evolution of the star.



Dense Matter in Compact Stars: Theoretical Developments and Observational Constraints, Page D. & Reddy S., 2006, Annu. Rev. Nucl. Part. Sci. 56, 327

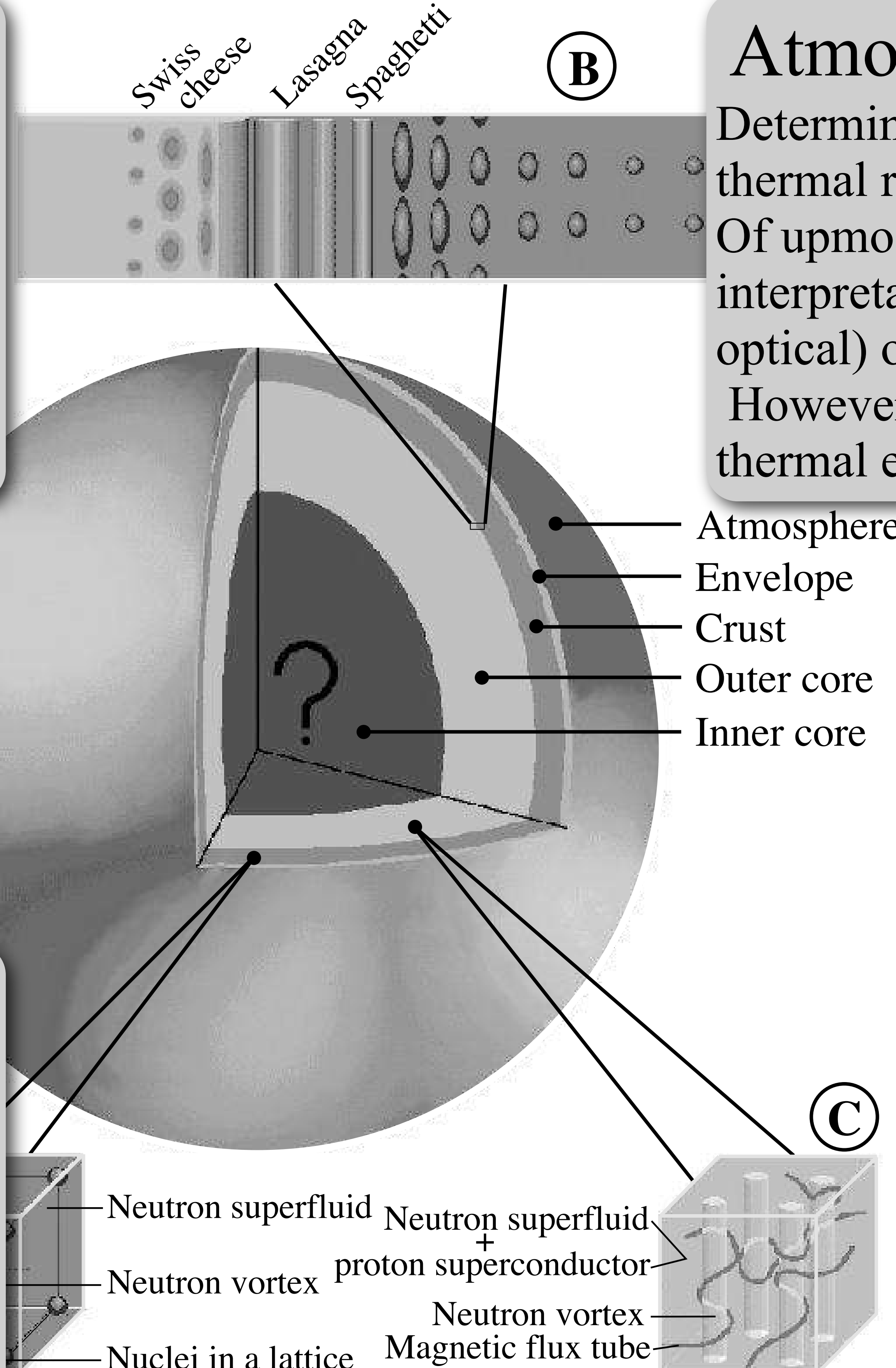
Envelope (100 m):

Contains a huge temperature gradient: it determines the relationship between T_{int} and T_e. Extremely important for the cooling, strongly affected by magnetic fields and the presence of “polluting” light elements.

Atmosphere (10 cm):

Determines the shape of the thermal radiation (the spectrum). Of upmost importance for interpretation of X-ray (and optical) observation.

However it as NO effect on the thermal evolution of the star.



Crust (1 km):

Little effect on the long term cooling. BUT: may contain heating sources (magnetic/ rotational, pycnonuclear under accretion). Its thermal time is important for very young star and for quasi-persistent accretion

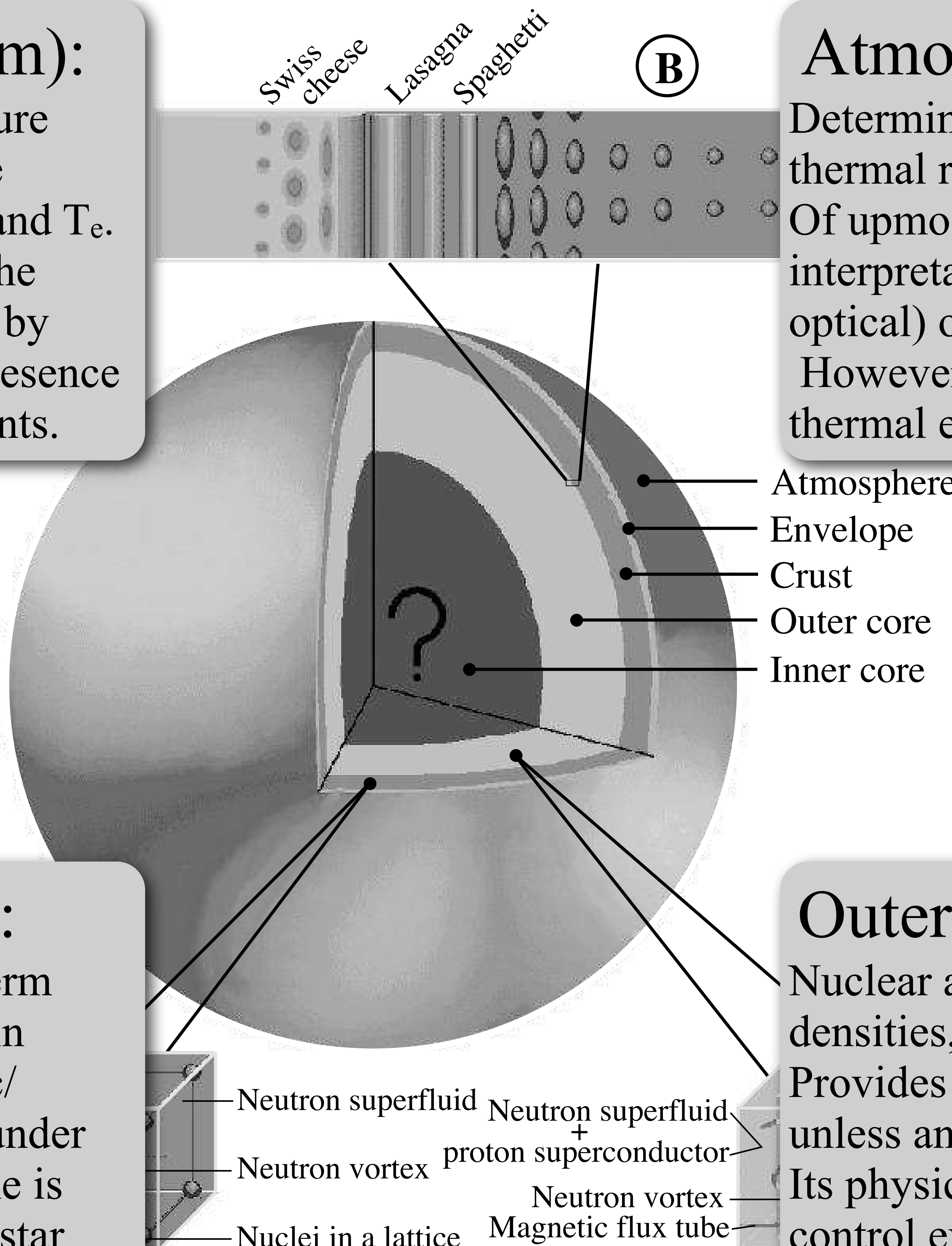
Dense Matter in Compact Stars: Theoretical Developments and Observational Constraints, Page D. & Reddy S., 2006, Annu. Rev. Nucl. Part. Sci. 56, 327

Envelope (100 m):

Contains a huge temperature gradient: it determines the relationship between T_{int} and T_e. Extremely important for the cooling, strongly affected by magnetic fields and the presence of “polluting” light elements.

Atmosphere (10 cm):

Determines the shape of the thermal radiation (the spectrum). Of upmost importance for interpretation of X-ray (and optical) observation.
However it as NO effect on the thermal evolution of the star.



Crust (1 km):

Little effect on the long term cooling. BUT: may contain heating sources (magnetic/ rotational, pycnonuclear under accretion). Its thermal time is important for very young star and for quasi-persistent accretion

Outer Core (10-x km):

Nuclear and supranuclear densities, containing n, p, e & μ. Provides about 90% of c_v and ε_ν unless an inner core is present. Its physics is basically under control except pairing T_c which is essentially unknown.

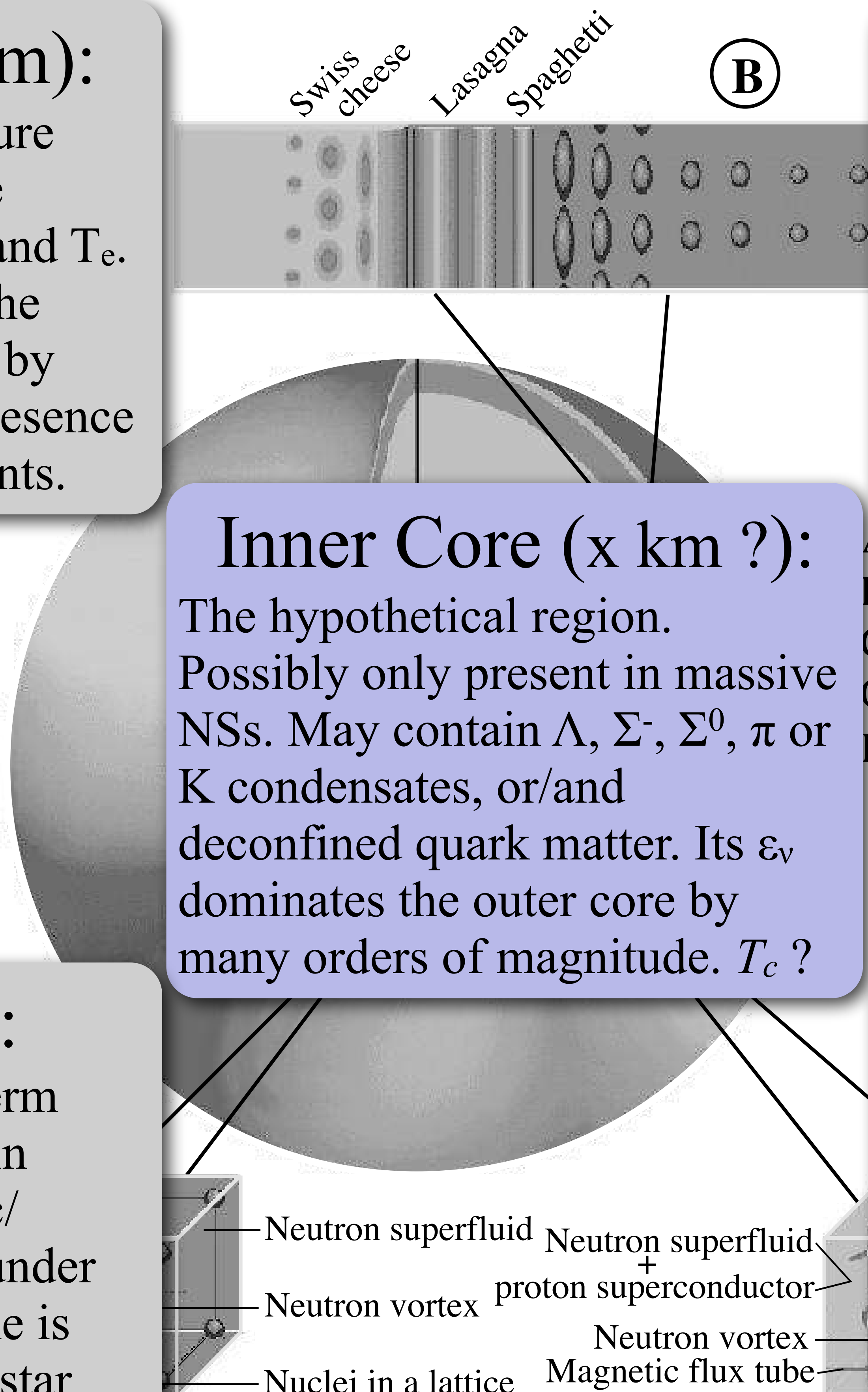
Dense Matter in Compact Stars: Theoretical Developments and Observational Constraints, Page D. & Reddy S., 2006, Annu. Rev. Nucl. Part. Sci. 56, 327

Envelope (100 m):

Contains a huge temperature gradient: it determines the relationship between T_{int} and T_e. Extremely important for the cooling, strongly affected by magnetic fields and the presence of “polluting” light elements.

Atmosphere (10 cm):

Determines the shape of the thermal radiation (the spectrum). Of upmost importance for interpretation of X-ray (and optical) observation.
However it as NO effect on the thermal evolution of the star.



Inner Core (x km ?):

The hypothetical region. Possibly only present in massive NSs. May contain Λ, Σ^-, Σ^0, π or K condensates, or/and deconfined quark matter. Its ϵ_ν dominates the outer core by many orders of magnitude. T_c ?

Crust (1 km):

Little effect on the long term cooling. BUT: may contain heating sources (magnetic/ rotational, pycnonuclear under accretion). Its thermal time is important for very young star and for quasi-persistent accretion

Outer Core (10-x km):

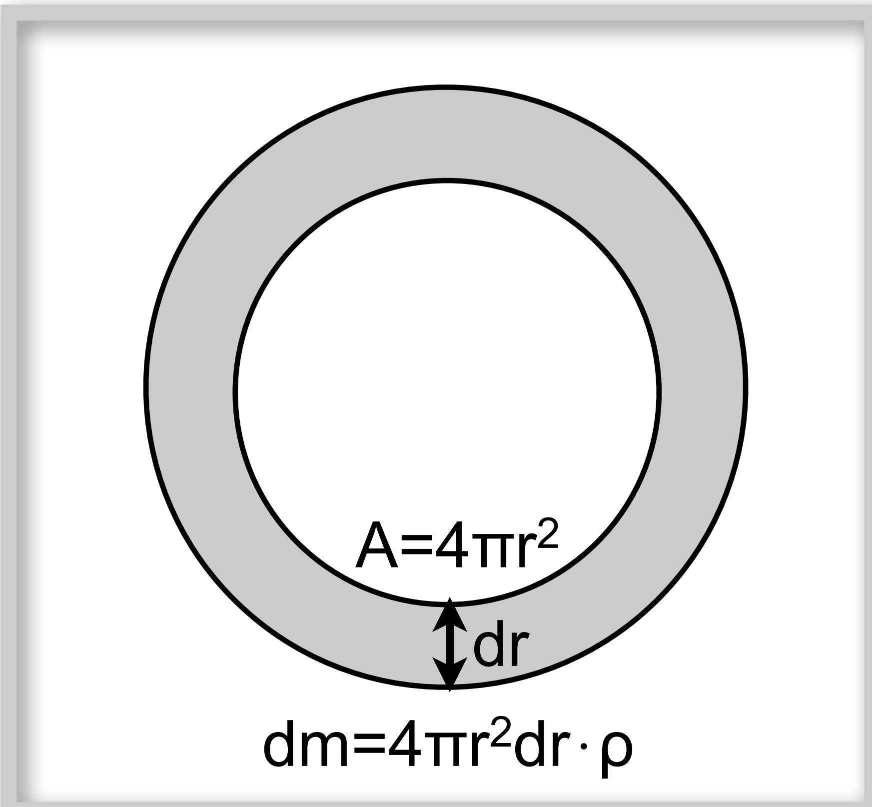
Nuclear and supranuclear densities, containing n, p, e & μ. Provides about 90% of c_v and ϵ_ν unless an inner core is present. Its physics is basically under control except pairing T_c which is essentially unknown.

Dense Matter in Compact Stars: Theoretical Developments and Observational Constraints, Page D. & Reddy S., 2006, Annu. Rev. Nucl. Part. Sci. 56, 327

Formulation of the Problem:

The Star Structure

The star structure: hydrostatic equilibrium

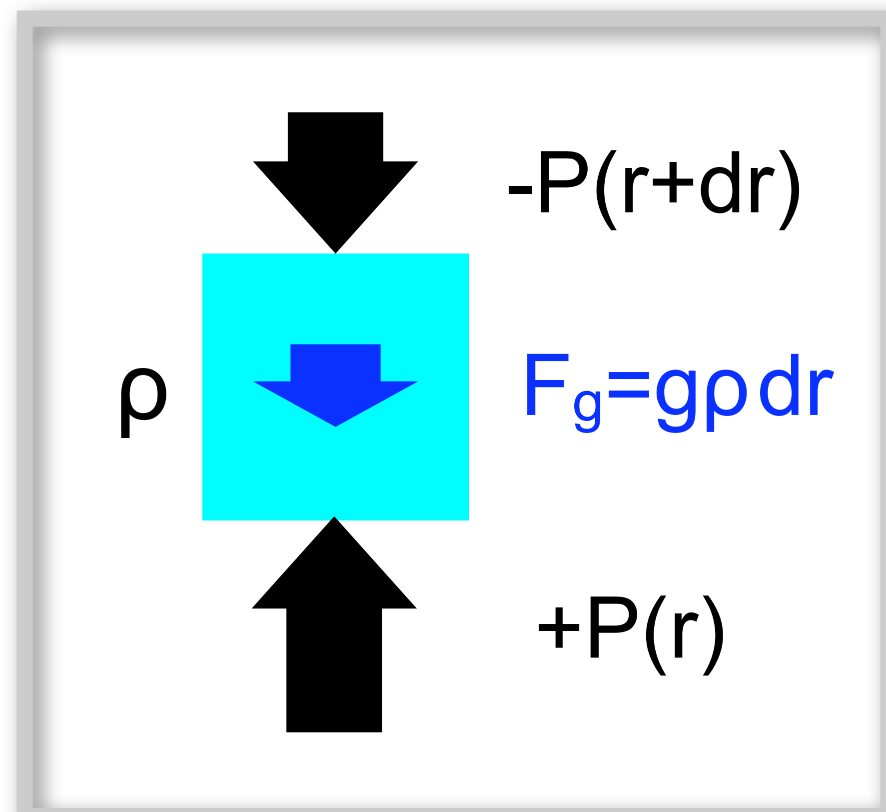


Mass equation: $\frac{dm}{dr} = 4\pi r^2$

Gravity: $g = -\frac{d\phi}{dr} = -\frac{Gm}{r^2}$

Hydrostatic equilibrium equation:

$$\frac{dP}{dr} = g\rho = -\frac{Gm\rho}{r^2}$$



The star's structure in GR: TOV

Newtonian formalism:		General relativistic formalism:
$\frac{dm}{dr} = 4\pi r^2 \rho$	Mass	$\frac{dm}{dr} = 4\pi r^2 \rho$
$\frac{d\phi}{dr} = \frac{Gm}{r^2}$	Gravitational potential	$\frac{d\Phi}{dr} = \frac{Gmc^2 + 4\pi G r^3 P}{c^4 r^2 (1 - 2Gm/c^2 r)}$
$\frac{dP}{dr} = -\rho \frac{d\phi}{dr} = -\frac{Gm\rho}{r^2}$	Hydrostatic equilibrium	$\frac{dP}{dr} = -(\rho c^2 + P)\frac{d\Phi}{dr} = -\frac{(\rho + P/c^2)(Gm + 4\pi G r^3 P/c^2)}{r^2(1 - 2Gm/c^2 r)}$

(Tolman- Oppenheimer-Volkoff equation)

$m = m_r$ is the mass (gravitational mass in GR) between 0 and r.

$\Phi = \Phi_r = \frac{1}{c^2}\phi$ is the gravitational potential

Notice: since P=P(ρ), no T dependence, we can calculate the star structure without solving for its thermal profile

The space-time metric inside the star

`NSCool` assumes a spherically symmetric star. The metric is

$$ds^2 = -e^{2\Phi(r)}\, c^2 dt^2 + \frac{dr^2}{1 - 2Gm(r)/rc^2} + r^2 d\Omega \qquad \text{where} \qquad d\Omega = \sin^2\theta d\phi^2 + d\theta^2$$

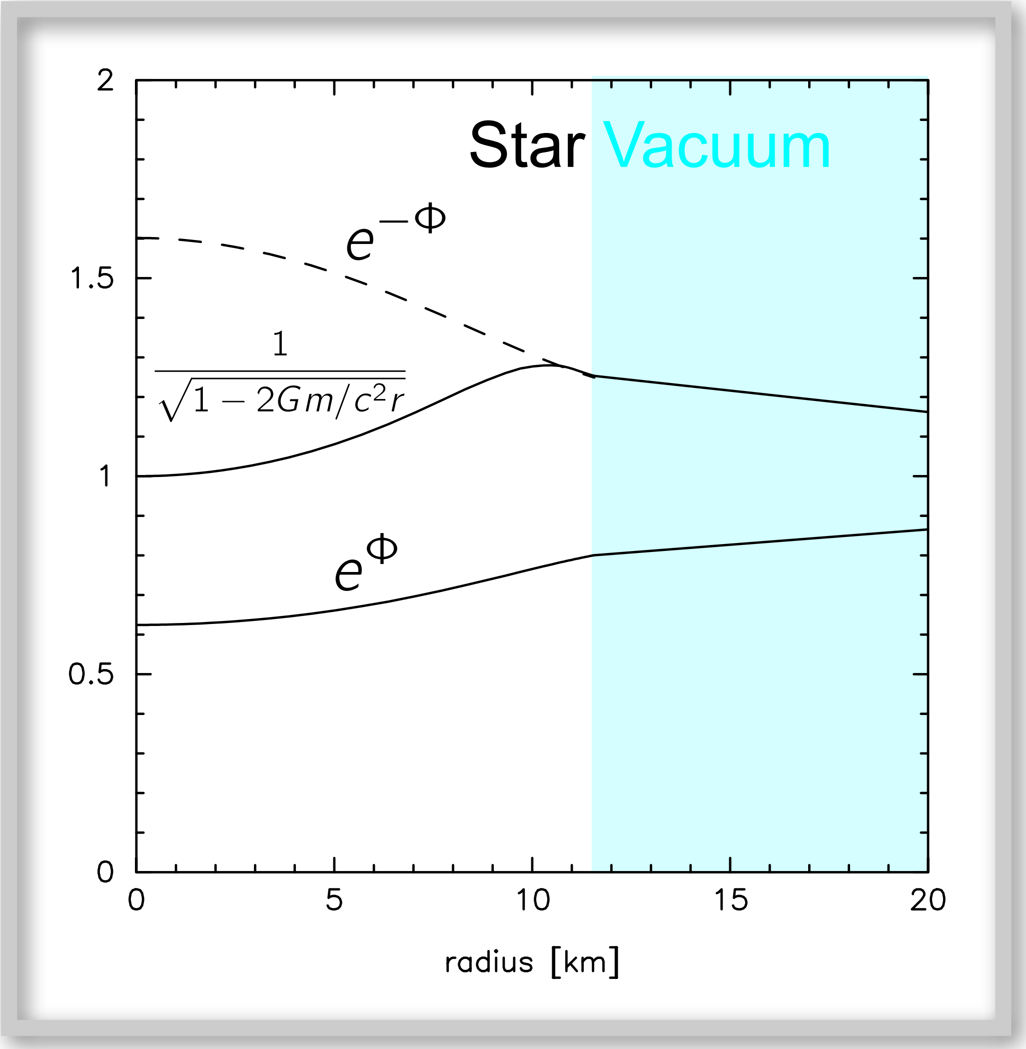
⇒ area of a sphere: $A_r = 4\pi r^2$

⇒ radial proper length: $dl = dr/\sqrt{1 - 2Gm(r)/rc^2}$

Physical radial length > radial coordinate r

⇒ proper time: $d\tau = e^{\Phi(r)} dt$

Time runs more slowly inside the NS



[Φ and m for a 1.4 $M_\odot$ star with the APR EOS]

At the surface of the star (r=R) it must match with the Schwarzschild solution for a gravitational mass M=m(R), so that:

$$e^{\Phi(R)} = \sqrt{1 - \frac{2GM}{c^2 R}}$$

Gravitational vs Baryonic Mass

Gravitational mass: $dm = 4\pi r^2 \rho \, dr$ $\qquad M = \int dm = \int 4\pi r^2 \rho \, dr$

Proper mass: $d\tilde{m} = 4\pi r^2 \rho \, dl$ (since *$4\pi r^2 dl$ = proper volume of a shell)*

Gravitational vs proper mass:

In the weak field limit:

$$dm = 4\pi r^2 \rho \, dr = 4\pi r^2 \rho \, dl \cdot \sqrt{1 - 2Gm/c^2 r} \simeq$$

$$4\pi r^2 \rho \, dl \cdot \left(1 - \frac{Gm}{c^2 r}\right) = d\tilde{m} - \frac{Gm \, d\tilde{m}}{r} \frac{1}{c^2}$$

More commonly one defines the baryon number: $da = 4\pi r^2 n_B \, dl$

(n_B is baryon number density, m_N the nucleon mass)

Baryonic mass: $M_B = m_N A$

where $A = \int da = \int 4\pi r^2 n_B \, dl$ is the total number of baryons in the star

Solving the TOV equation(s)

The four equations (mass, grav. potential, hydro equilibrium and baryon number) are easily solved by a 4th order Runge-Kutta integration. Such a code is provided, in the directory `TOV`, and its use is described in `NSCool_Guide_TOV`.

Integration starts at r=0 with the initial conditions:

$$m(r=0)=0 \quad ; \quad n_B(r=0)=n_c \quad ; \quad \phi(r=0)=0$$

and, using the EOS the corresponding ρ_c and P_c are also fixed.

Integrations stops when *P=0*: that's the surface ! (But ρ may be non-zero at the surface.) The value of *r* is thus the star's radius *R*, and *m(r=R)=M*.

Since ϕ only appears as its derivative, one can shift it ($\Phi \rightarrow \Phi + \Phi_0$) so that it matches the vacuum Schwarzschild solution at the surface:

$$e^{\Phi(R)} = \sqrt{1 - \frac{2GM}{c^2 R}}$$

- The result of the integration is a table of r ; n_B ; ρ ; P ; m ; Φ ; a. Samples of such files are in the subdirectory `TOV/Profile`.
- By varying ρ_c one can generate a family of stars and so an M-ρ_c *(or an* M-R) curve. Samples of such tables are in the subdirectory `TOV/Production`.

Formulation of the Problem:

The Star Evolution

Energy transport

Fick's law: $F = -\lambda \nabla T$ for the heat flux **F** (in erg cm^{-2} s^{-1})
where λ = thermal conductivity (often unfortunately written as κ)

In a star (with spherical symmetry):

$$\frac{1}{4\pi r^2} L(r) = F(r) = -\lambda \frac{dT}{dr}$$

where *L(r)* is the (diffusive) luminosity at radius *r* (in erg s^{-1}).

The thermal conductivity *λ* is related to the opacity κ (for photons) by: $\lambda = \frac{16\sigma_{SB} T^3}{3\kappa\rho}$

General Relativistic version:

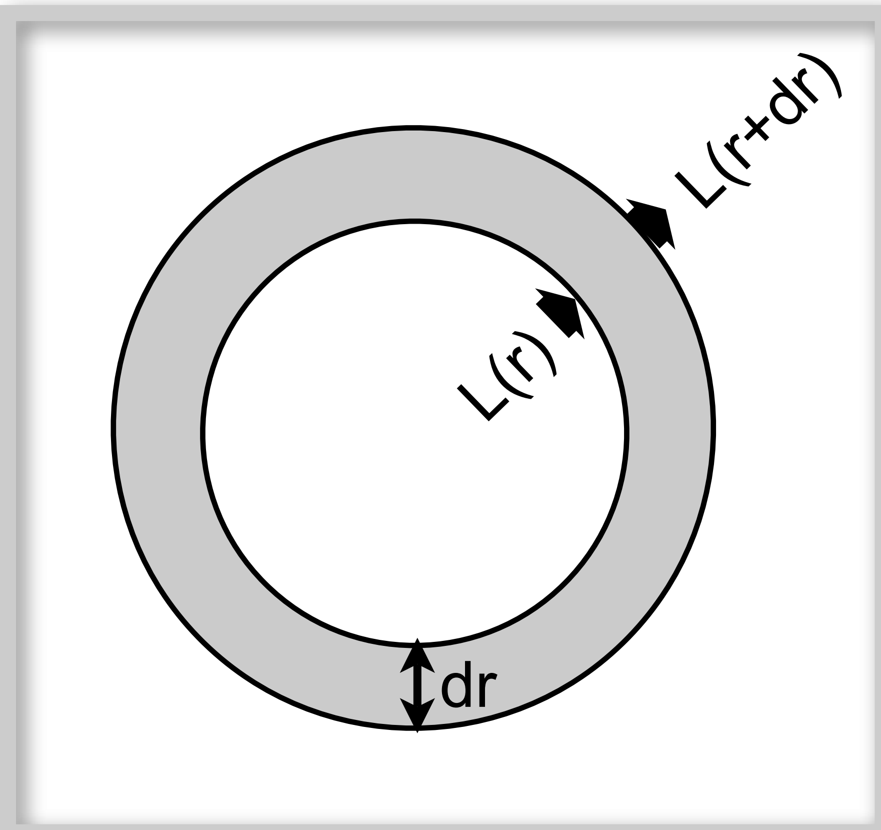
Energy transport:

$$\frac{d(Te^{\Phi})}{dr} = -\frac{1}{\lambda} \cdot \frac{Le^{\Phi}}{4\pi r^2 \sqrt{1 - 2Gm/c^2 r}}$$

Notice: energy has to be red-shifted: Te^{ϕ} and Le^{ϕ}

Thermal equilibrium **("isothermal" star)**: $L = 0 \longrightarrow Te^{\phi} = \text{constant}$

Energy balance



$$L(r+dr) = L(r) + \left[Q_h - Q_\nu - \frac{dU}{dt} \right] \times dV$$

where

U = internal energy per unit volume, and

Q_ν = neutrino emissivity

Q_h = heating rate,

both per unit volume per unit time.

(it is assumed that neutrinos leave the star).

U can be expressed in terms of the specific heat C_v (which for degenerate matter is the same as C_P) per unit volume:

$$\frac{dU}{dt} = \frac{dU}{dT} \cdot \frac{dT}{dt} = C_v \cdot \frac{dT}{dt}$$

Energy balance

$$L(r+dr) = L(r) + \left[Q_h - Q_\nu - \frac{dU}{dt}\right] \times dV$$

General Relativistic version:

Energy balance:

$$\frac{d(Le^{2\Phi})}{dr} = -\frac{4\pi r^2 e^{\Phi}}{\sqrt{1-2Gm/c^2 r}}\left(\frac{dU}{dt} + e^{\Phi}(Q_\nu - Q_h)\right)$$

or: $$d(Le^{2\Phi}) = -\left(\frac{dU}{d\tau} + (Q_\nu - Q_h)\right) e^{2\Phi} \times 4\pi r^2\, dl$$

➥ *L* must be red-shifted twice: only red-shifted energy is conserved and time runs differently at *r* and *r+dr*.

➥ $\frac{dU}{d\tau},\ Q_h,\ \text{and}\ Q_\nu$ are proper energy per proper volume and proper time and must be adjusted

Solving these equations in NSCool

The problem to be solved

The equations to be solved are described in the NSCool_Guide_1 Introduction. They are:

1) Structure of the star: the TOV equations.
2) Thermal evolution of the star.

How to use the TOV solver is described in NSCool_Guide_3_TOV. Meanwhile, several pre-built stars are available in the directory `TOV/Profile`.

For the thermal evolution equations, the star is cut at an outer boundary, with radius r_b and density ρ_b (typically $\rho_b = 10^{10}$ gm cm^{-3}): at $\rho > \rho_b$ matter is strongly degenerate and thus the structure of the star does not change with time:

The star's structure is calculated before the cooling and not modified thereafter.

(Almost: `NSCool` allows for small density changes in the outer part of the star, if required)

Only the energy balance and transport equations are solved as a function of time:

- two first order partial differential equations to get $L(r,t)$ and $T(r,t)$ with
- an initial L and T profile: $L(r,t=0)$ and $T(r,t=0)$
- two boundary conditions, at $r=0$ and $r=r_b$.

Note: the heat transport is a diffusion equation and numerically unstable if treated improperly. Numerical stability is achieved using an implicit scheme ("Henyey scheme") similar to the textbook Crank-Nicholson.

Rewriting the thermal evolution equations

The equations to solve:

Energy balance

$$\frac{d(Le^{2\Phi})}{dr} = -\frac{4\pi r^2 e^{\Phi}}{\sqrt{1-2Gm/c^2r}} \left(C_v \frac{dT}{dt} + e^{\Phi}(Q_\nu - Q_h) \right)$$

Energy transport

$$\frac{d(Te^{\Phi})}{dr} = -\frac{1}{\lambda} \cdot \frac{Le^{\Phi}}{4\pi r^2 \sqrt{1-2Gm/c^2r}}$$

Use red-shifted functions: $\mathcal{T} \equiv e^{\Phi} T$ and $\mathcal{L} \equiv e^{2\Phi} L$

and the Lagrangian coordinate *a* (baryon number) $da = 4\pi r^2 dl\, n_B = \frac{4\pi r^2 n_B\, dr}{\sqrt{1-2Gm/c^2r}}$

to get: $$\frac{d\mathcal{L}}{da} = -\frac{C_v}{n_B}\frac{d\mathcal{T}}{dt} - e^{2\Phi}\frac{Q_\nu - Q_h}{n_B} \quad \text{or} \quad \frac{d\mathcal{T}}{dt} = -e^{2\Phi}\frac{Q_\nu - Q_h}{C_v} - \frac{n_B}{C_v}\frac{d\mathcal{L}}{da}$$

and: $$\frac{d\mathcal{T}}{da} = -\frac{1}{\lambda}\frac{\mathcal{L}}{(4\pi r^2)^2 n_B\, e^{\Phi}} \quad \text{or} \quad \mathcal{L} = -\lambda\, (4\pi r^2)^2 n_B\, e^{\Phi}\, \frac{d\mathcal{T}}{da}$$

which we write as: $$\frac{d\mathcal{T}}{dt} = F\left(\mathcal{T}, \frac{d\mathcal{L}}{da}\right) \quad \text{and} \quad \mathcal{L} = G\left(\mathcal{T}, \frac{d\mathcal{T}}{da}\right)$$

(the $\mathcal{T}$ dependance of *F* and *G* comes from Q_v, Q_h, C_v, and λ)

Finite differencing the equations

For finite differencing these equations one divides the star into shells, at radii $r_0=0$, r_1, ... , r_i , ... r_{imax}. *L,* being a flux, is defined at the shell interfaces while T is understood as the average in the interior of each shell: it is common to write then L_i and $T_{i+½}$ to emphasize this.

Since fortran does not like loop indices with half integer values I used:

$\mathcal{L}$ is defined at i = 0, 2, 4, ... , i_{max}-1

$\mathcal{T}$ is defined at i = 1, 3, 5, ... , i_{max}

$$\frac{d\mathcal{T}}{dt} = F\left(\mathcal{T}, \frac{d\mathcal{L}}{da}\right) \longrightarrow \frac{d\mathcal{T}_i}{dt} = F\left(\mathcal{T}_i, \left.\frac{d\mathcal{L}}{da}\right|_i\right) \quad \text{with} \quad \left.\frac{d\mathcal{L}}{da}\right|_i = \frac{\mathcal{L}_{i+1} - \mathcal{L}_{i-1}}{da_{i-1} + da_i}$$

for i = 1, 3, 5, ...

$$\mathcal{L} = G\left(\mathcal{T}, \frac{d\mathcal{T}}{da}\right) \longrightarrow \mathcal{L}_i = G\left(\mathcal{T}|_i\,, \left.\frac{d\mathcal{T}}{da}\right|_i\right) \quad \text{with} \quad \mathcal{T}|_i = \frac{\mathcal{T}_{i+1} + \mathcal{T}_{i-1}}{2} \quad \text{and} \quad \left.\frac{d\mathcal{T}}{da}\right|_i = \frac{\mathcal{T}_{i+1} - \mathcal{T}_{i-1}}{da_{i-1} + da_i}$$

for i = 2, 4, 6, ...

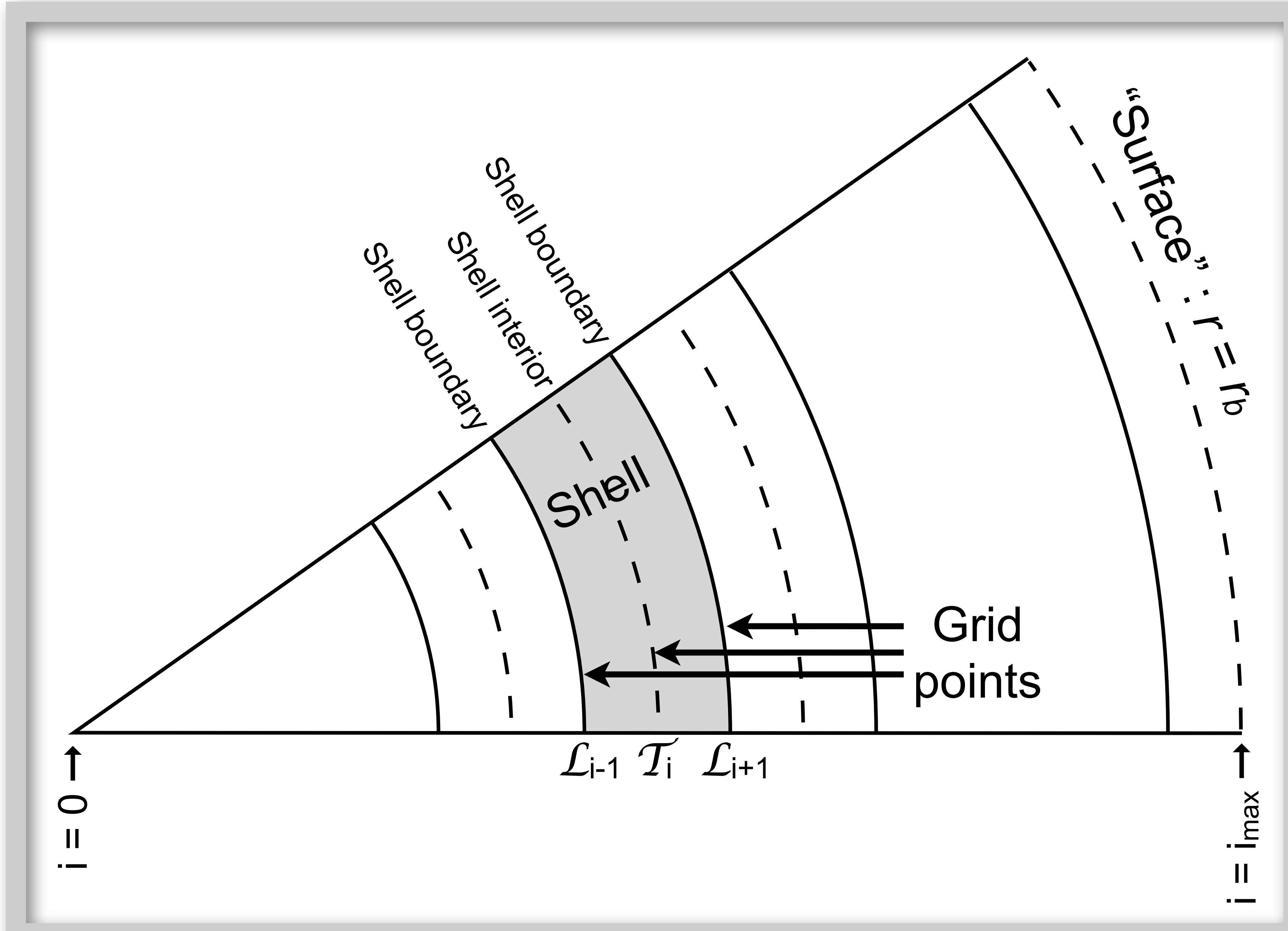
where da_i is the number of baryons between r_{i-1} and r_i

"Zoning" the star

The star is discretized, at radii r_i, i=0, 1, 2, ...,i_{max} with $r_{i=0} = 0$ and $r_{i=i_{max}} = r_b$ (the outer radius in the simulation)

$\mathcal{L}$ is defined at the boundaries between shells, i.e.,
i = 0, 2, 4, ... , i_{max}-1

$\mathcal{T}$ is defined inside the sheels, i.e.,
i = 1, 3, 5, ... , i_{max}



Stepping forward in time

Assuming we know the profiles of $\mathcal{T}$ and $\mathcal{L}$ at time *t*: $\mathcal{T}^{\rm old}$ and $\mathcal{L}^{\rm old}$

we can write for $\mathcal{T}$ and $\mathcal{L}$ at time *t'=t+dt*:

Explicit scheme

$$\frac{d\mathcal{T}}{dt} = F\left(\mathcal{T}, \frac{d\mathcal{L}}{da}\right) \longrightarrow \mathcal{T} = \mathcal{T}^{\rm old} + dt \cdot F\left(\mathcal{T}^{\rm old}, \frac{d\mathcal{L}^{\rm old}}{da}\right)$$

$$\mathcal{L} = G\left(\mathcal{T}, \frac{d\mathcal{T}}{da}\right) \longrightarrow \mathcal{L} = G\left(\mathcal{T}^{\rm old}, \frac{d\mathcal{T}^{\rm old}}{da}\right)$$

this is very easy to integrate BUT:
it is numerically unstable unless *dt* is very small (Courant *dixit*)

Better: evaluate *F* and *G* at the new values of $\mathcal{T}$ and $\mathcal{L}$:

Implicit scheme

$$\frac{d\mathcal{T}}{dt} = F\left(\mathcal{T}, \frac{d\mathcal{L}}{da}\right) \longrightarrow \mathcal{T} = \mathcal{T}^{\rm old} + dt \cdot F\left(\mathcal{T}, \frac{d\mathcal{L}}{da}\right)$$

$$\mathcal{L} = G\left(\mathcal{T}, \frac{d\mathcal{T}}{da}\right) \longrightarrow \mathcal{L} = G\left(\mathcal{T}, \frac{d\mathcal{T}}{da}\right)$$

this is numerically stable (and allows large *dt*) BUT:
extracting the new $\mathcal{T}$ and $\mathcal{L}$ is tough
(particularly $\mathcal{T}$ because it is inside Q_v, Q_h, C_v, and λ)

Solving the implicit equations by iterations

Assuming we know the profiles of $\mathcal{T}$ and $\mathcal{L}$ at time *t*: $\mathcal{T}^{old}$ and $\mathcal{L}^{old}$
we can find the new $\mathcal{T}$ and $\mathcal{L}$ at time *t'*=*t*+*dt* by successive approximations
$(\mathcal{T}^{(0)}, \mathcal{L}^{(0)}) \to (\mathcal{T}^{(1)}, \mathcal{L}^{(1)}) \to (\mathcal{T}^{(2)}, \mathcal{L}^{(2)}) \to (\mathcal{T}^{(3)}, \mathcal{L}^{(3)}) \to \ldots$

As an initial guess for $(\mathcal{T}^{(0)}, \mathcal{L}^{(0)})$ one can take $(\mathcal{T}^{(0)}, \mathcal{L}^{(0)}) = (\mathcal{T}^{old}, \mathcal{L}^{old})$
or extrapolate from $(\mathcal{T}^{old}, \mathcal{L}^{old})$ and the previous values $(\mathcal{T}^{older}, \mathcal{L}^{older})$.

Evaluate the functions F and G with $\mathcal{T}_i^{(k)}$ and $\mathcal{L}_i^{(k)}$ to obtain $\mathcal{T}_i^{(k+1)}$ and $\mathcal{L}_i^{(k+1)}$:

$$\mathcal{T}_i^{(k+1)} = \mathcal{T}_i^{old} + dt \cdot F\left(\mathcal{T}_i^{(k)}, \left.\frac{d\mathcal{L}}{da}\right|_i^{(k)}\right) \qquad \mathcal{L}_i^{(k+1)} = G\left(\mathcal{T}\Big|_i^{(k)}, \left.\frac{d\mathcal{T}}{da}\right|_i^{(k)}\right)$$

then plug back $\mathcal{T}_i^{(k+1)}$ and $\mathcal{L}_i^{(k+1)}$ into F and G to obtain $\mathcal{T}_i^{(k+2)}$ and $\mathcal{L}_i^{(k+2)}$
and so on until some K when $\mathcal{T}_i^{(K+1)} \cong \mathcal{T}_i^{(K)}$ and $\mathcal{L}_i^{(K+1)} \cong \mathcal{L}_i^{(K)}$
(All $\mathcal{T}_i^{(K)}$ and $\mathcal{L}_i^{(K)}$ are successive approximations *at the same time t'*. $\mathcal{T}_i^{old}$ does not change, it is at time t !)

As long as the initial guess $(\mathcal{T}^{(0)}, \mathcal{L}^{(0)})$ is not too far from the solution
the method will converge to the solution (maybe in 10 iterations ?)

Improvement: the Henyey scheme

Instead of using brute force iterations, the Henyey scheme use the Newton-Raphson method for solving multi-dimensional equations.

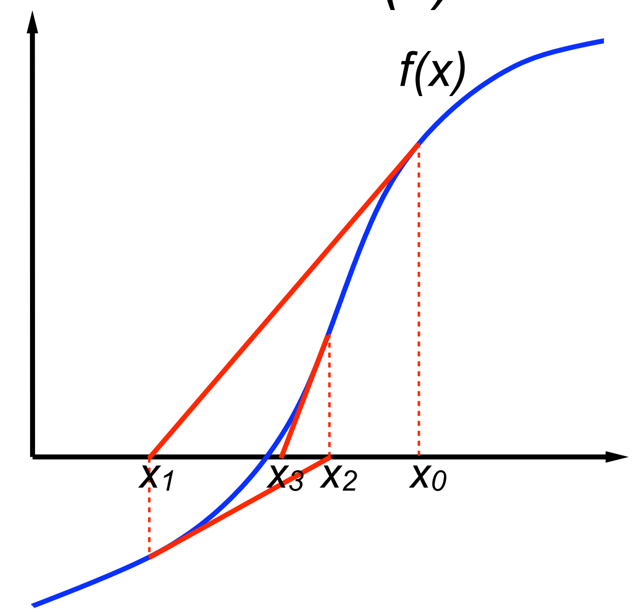
Write the equations as:

$$\begin{cases} \mathcal{T} - \mathcal{T}^{\text{old}} - dt \cdot F\left(\mathcal{T}, \frac{d\mathcal{L}}{da}\right) = 0 \\ \mathcal{L} - G\left(\mathcal{T}, \frac{d\mathcal{T}}{da}\right) = 0 \end{cases}$$

or, in N dimensional notation:

$$\Phi(X) = 0 \quad \text{with} \quad X = \begin{pmatrix} \mathcal{L}_0 \\ \mathcal{T}_1 \\ \mathcal{L}_2 \\ \mathcal{T}_3 \\ . \\ . \end{pmatrix} \quad \text{and} \quad \Phi(X) = \begin{pmatrix} \Phi_0(X) \\ \Phi_1(X) \\ \Phi_2(X) \\ \Phi_3(X) \\ . \\ . \end{pmatrix}$$

The Newton method to solve *f(x)=0*



$$f(x_{k+1}) = 0 \quad \leftrightarrow$$

$$f(x_k) + f'(x_k) \cdot (x_{k+1} - x_k) = 0$$

$$\Longrightarrow x_{k+1} = x_k - [f'(x_k)]^{-1} \cdot f(x_k)$$

and the Newton-Raphson iteration procedure is: $X^{(k+1)} = X^{(k)} - [D\Phi(X^{(k)})]^{-1} \cdot \Phi(X^{(k)})$

where $[D\Phi(X)]$ is the NxN derivative matrix of $\Phi(X)$ and $[D\Phi(X)]^{-1}$ the inverse matrix.

This involves calculating T derivatives of Q_v, Q_h, C_v, and λ and inverting a large matrix. Fortunately this matrix is tri-diagonal and its inversion is straightforward !

One still have to preform iterations but the convergence can be much faster than brute force.

Checking for iteration convergence and time step control

The Newton-Raphson iterations go as:

$\mathcal{T}_i^{(k)} \rightarrow \mathcal{T}_i^{(k+1)} = \mathcal{T}_i^{(k)} + \delta\mathcal{T}_i^{(k)}$ [i=1, 3, 5, ...]

$\mathcal{L}_i^{(k)} \rightarrow \mathcal{L}_i^{(k+1)} = \mathcal{L}_i^{(k)} + \delta\mathcal{L}_i^{(k)}$ [i=0, 2, 4, ...]

Convergence will be considered to have been achieved when

$$\mathrm{Max}_{i=1,3,5,...}\left(\frac{\delta\mathcal{T}_i^{(k)}}{\mathcal{T}_i^{(k)}}\right) < \epsilon_T \quad \text{and} \quad \mathrm{Max}_{i=0,2,4,...}\left(\frac{\delta\mathcal{L}_i^{(k)}}{\mathcal{L}_i^{(k)}}\right) < \epsilon_L$$

Values of ε_T and ε_L of the order of 10^{-10} can be reached in 4 - 6 iterations.

However, if $\mathcal{T}_i^{(0)}$ and/or $\mathcal{L}_i^{(0)}$ are too far away from the solution, iterations go on forever: the loop is exited, the time step *dt* is shortened and the iteration procedure restarted.

(It is not unusual to see *dt* being cut many times, e.g., when a phase transition (superfluidity/superconductivity) occurs at some point in the star. Sometimes things go real bad (dt → almost zero): "Ctrl-C" is the only solution, and figure out what's happening.)

Time step control: at every new time step *dt* is increased: *dt* → *dt (1+α)* (*α ~ 0.2*) but:

- if Newton-Raphson converged in << 5 steps a larger α is chosen
- if Newton-Raphson needed > 10 steps to converge a smaller α is chosen
- if $\mathcal{T}$ and/or $\mathcal{L}$ changed too much (from $\mathcal{T}^{old}$ and/or $\mathcal{L}^{old}$) a smaller α is chosen, while if they changed ways too much, the time step is recalculated with a smaller *dt*.

The boundary conditions

Inner boundary condition: $L(r=0) = 0$ or $\mathcal{L}_{i=0} = 0$

This is easily implemented by initially starting with $\mathcal{L}_{i=0}^{(k=0)} = 0$ and imposing $\delta\mathcal{L}_{i=0}^{(k)} = 0$ at every iteration.

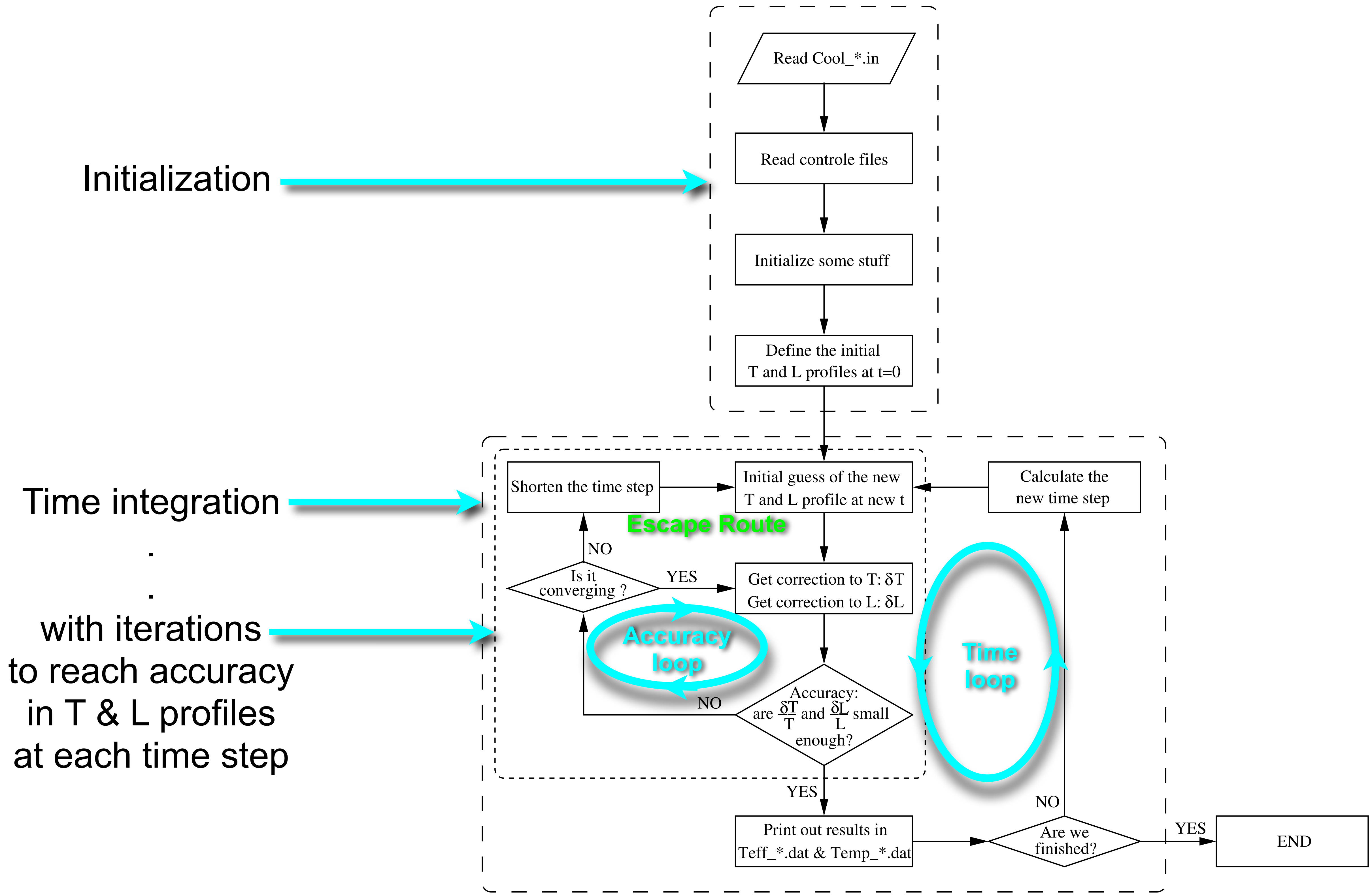
Outer boundary condition (see NSCool_Guide_1_Introduction):

It is (at $r = r_b$): $L(r_b) = 4\pi R^2 \sigma_{SB} [T_e(T_b)]^4$ with $T_b \equiv T(r_b)$

where (in present notations): $L(r_b) = e^{-2\Phi(i_{max}-1)} \mathcal{L}(i_{max}-1)$ and $T(r_b) = e^{-\Phi(i_{max})} \mathcal{T}(i_{max})$ and $T_e(T_b)$ is a function (a "T_e-T_b" relationship) obtained from some envelope model.

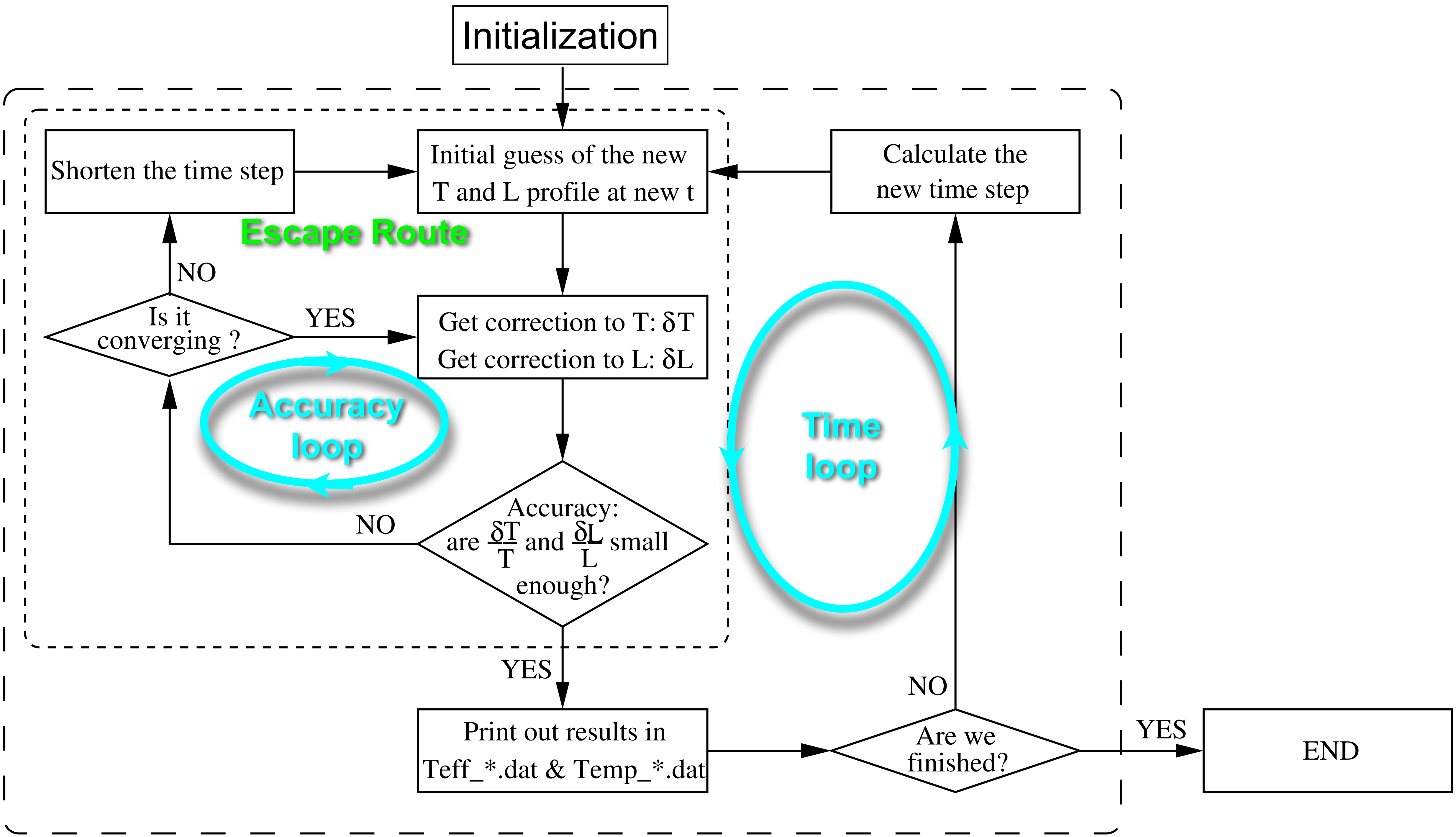
This is implemented as part of the inversion of the matrix $[D\Phi(X)]$

Flow diagram of NSCool



Notice: NSCool contains an extra "model loop" to run several cooling models from the same Cool_*.in file.

Flow diagram of NSCool (bigger)




"That's all Folks!"

NSCool web site

http://www.astroscu.unam.mx/neutrones/NSCool/