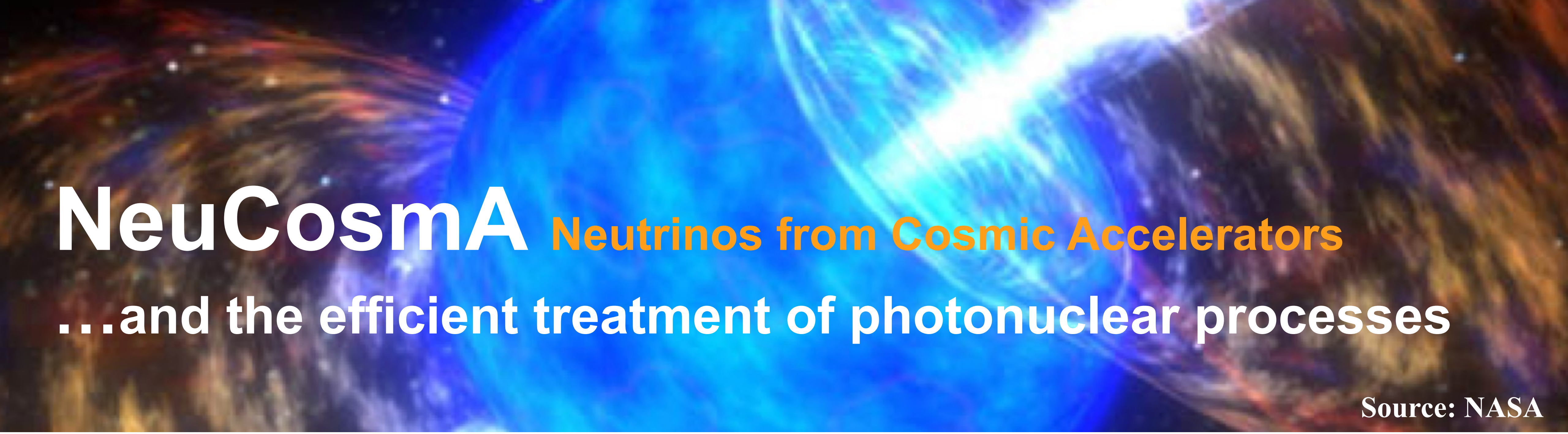
NeuCosmA Neutrinos from Cosmic Accelerators

…and the efficient treatment of photonuclear processes



Source: NASA

Winter, Walter
DESY, Zeuthen, Germany

Astrophysics Workshop on Numerical Multimessenger Modeling

Uni Bochum, Germany
Feb. 27-March 1, 2023

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

DESY

Contents

Numerical solution of PDE systems

- Time-dependent PDE system, one PDE per particle species *i*

$$\frac{\partial N_i}{\partial t} = \frac{\partial}{\partial E}\left(-b(E)N_i(E)\right) - \frac{N_i(E)}{t_{\mathrm{esc}}} + Q(E)$$

Cooling (continuous) | Escape | Injection

“radiation processes”

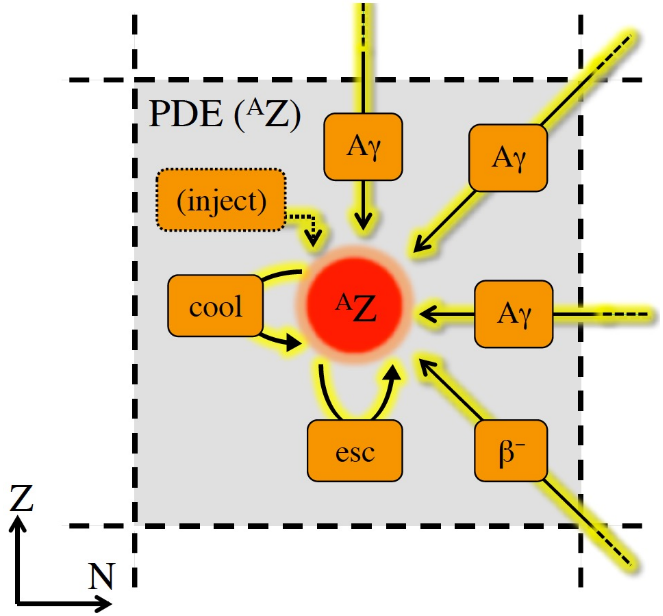
$b(E) = -E\, t^{-1}_{\mathrm{loss}}$
$Q(E,t)$ [$\mathrm{GeV}^{-1}\ \mathrm{cm}^{-3}\ \mathrm{s}^{-1}$]
$N(E,t)$ [$\mathrm{GeV}^{-1}\ \mathrm{cm}^{-3}$] particle spectrum including spectral effects

- Injection: species *i* from acceleration zone, and from other species *j*:

$$Q(E) = Q_i(E) + Q_{ji}(E)$$

$$Q_{ji}(E_i) = \int dE_j\, N_j(E_j)\, \Gamma_j^{\mathrm{IT}}(E_j)\, \frac{dn^{\mathrm{IT}}_{j\to i}}{dE_i}(E_j, E_i)$$

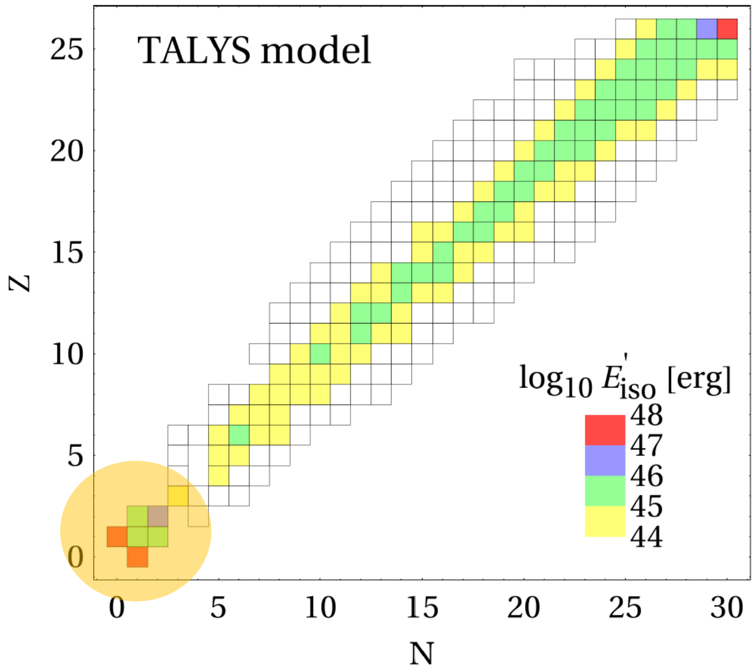
Density other species	Inter-action rate	Re-distribution function +secondary multiplicity



Computation of the nuclear cascade (GRB) with NeuCosmA

Disintegration in a GRB

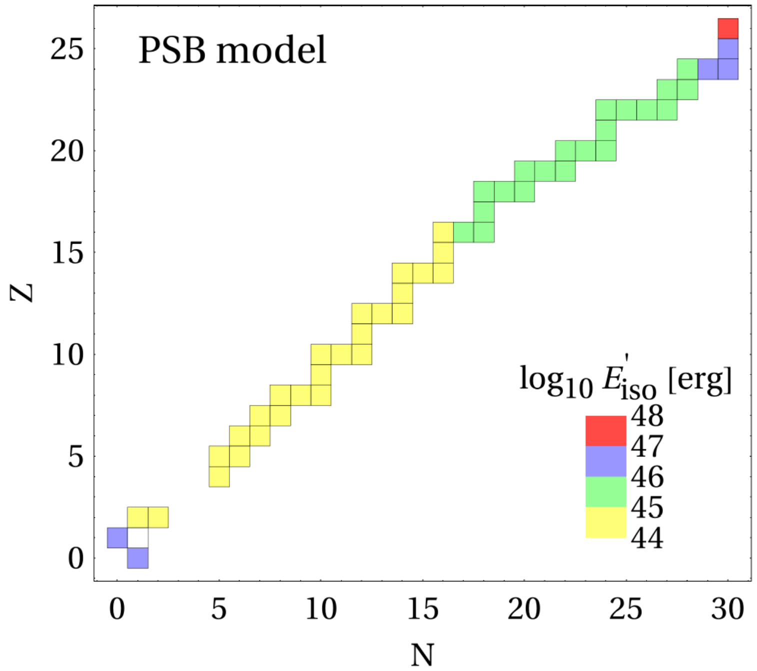
- Population of many isotopes by nuclear cascade:



- Disintegration of ^{56}Fe ***within*** a GRB shell c ($L_\gamma=10^{52}$ erg/s)

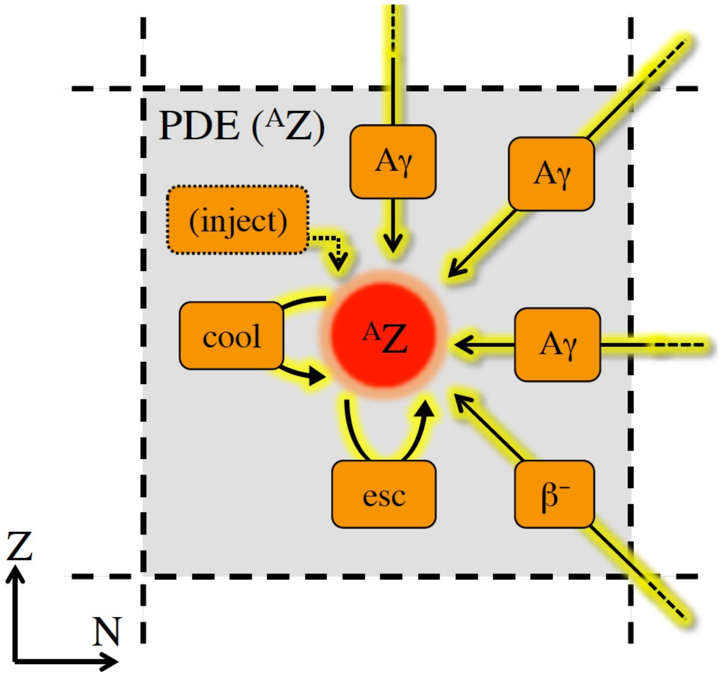
PSB disintegration model

- Puget-Stecker-Bredekamp model: one isobar approxim.



- Lighter elements less populated than for TALYS, PEANUT

Each box (isotope) corresponds to one PDE:



Boncioli, Fedynitch, Winter, Scientific Reports 7 (2017) 4882

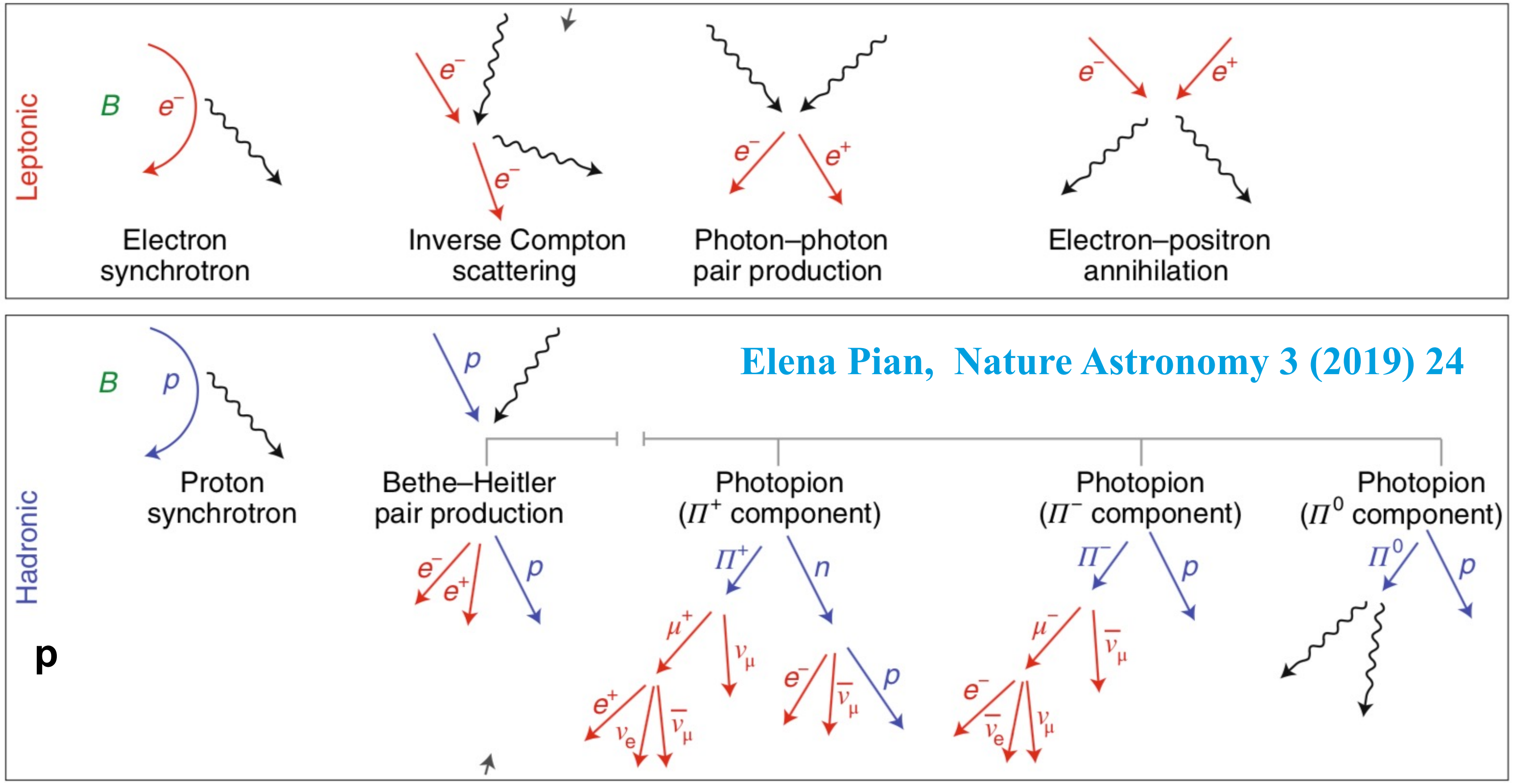
Radiation processes

Hadronic (nuclei A)

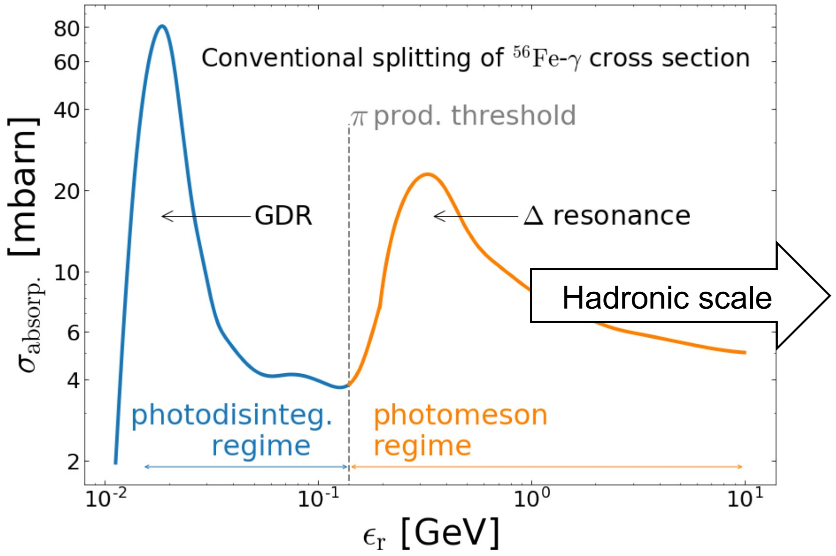
- **QED scale**: $\varepsilon_r > 2\,m_e c^2 \sim 1$ MeV
  Aγ ➔ Ae^+e^- pair production
- **Nuclear scale**: $\varepsilon_r > 8$ MeV
  Nuclear **photo-disintegration**
  (produces unstable isotopes)
- **Mesonic/QCD scale**:
  $\varepsilon_r > 140$ MeV
  Baryonic resonances,
  photo-meson production
  (produces neutrinos)
- **Hadronic scale**: $\varepsilon_r > 1$ GeV
  Hadronic structure matters

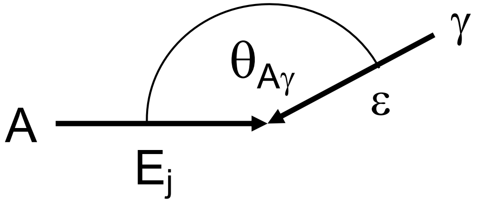
- Beta decays, spontaneous nucleon emissions, spallation, de-excitation, ...

Morejon, Fedynitch, Boncioli, Biehl, Winter, JCAP 11 (2019) 007



Elena Pian, Nature Astronomy 3 (2019) 24





$$\epsilon_r = \frac{E_j\,\varepsilon}{m_A}(1 - \cos\theta_{A\gamma})$$

$$s(\epsilon_r) = m_A^2 + 2\,m_A\,\epsilon_r$$

The NeuCosmA code

- C99 library with application software examples. Proprietary code.
- Idea: **balance precision and performance**. High flexibility regarding processes and species.
- Fully time-dependent PDE solver, direct steady-state solvers also available (deprecated) (Crank Nicolson scheme with single energy grid, default log E - log (E^2 N) – lin. t parameterization, super-robust wrt stiffness)
- Focused on hadronic processes and pion decay chain, kaon production
- Cooling effects of secondary pions, muons, kaons (time-dependent or steady state solvers)
- Acceleration (1^{st} order), in principle, possible, but numerically not well tested
- Global radiation model (no spatial dimensions)
- Can handle thousands of nuclear isotopes and ten thousands of channels (disintegration, beta decays etc). Different interaction and disintegration models (e.g. TALYS, PSB) implemented
- Photon fields as external input, i.e., test particle approach wrt photons: But: Exploratory project coupling NeuCosmA with AM^3
- Lot of flexibility (not only switches, but structure of system, interaction models, channels to be used, radiation processes etc. have to be specified in a modular way), but as a consequence application software more involved

NeuCosmA library
`nco_time.c`: time-dep. PDE solver
`nco_photo.c`: Aγ interactions
`nco_proto.c`: Ap interactions
… (many more)

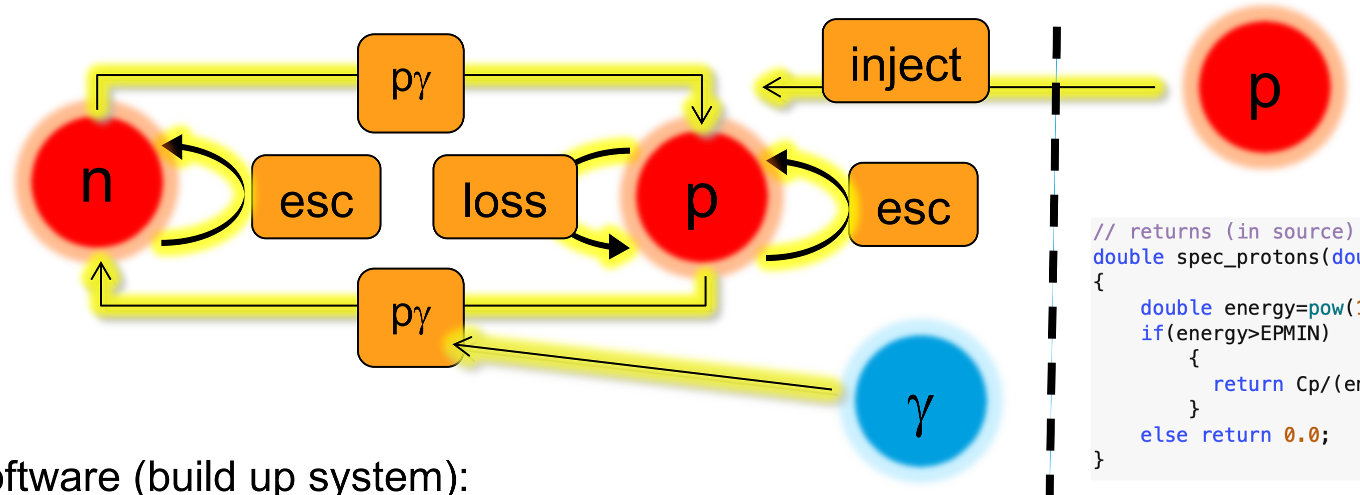
↓

Application software
(problem/source-specific; C or Python)

↓

Output (text files or Jupyter notebook)

Code excerpt example: Coupled proton-neutron system




```
// returns (in source) proton spectrum in 1/GeV 1/cm^3 1/s
double spec_protons(double x, double t, void* s)
{
    double energy=pow(10.0,x);
    if(energy>EPMIN)
        {
          return Cp/(energy*energy)*exp(-pow(energy/maxprotonenergy,1.0));
        }
    else return 0.0;
}
```

From the application software (build up system):

Hook to a function

```
int photons=ncoAddSpeciesF(NCO_PHOTON,NCO_VARIABLE,spec_photons);
// static dummy for external input, here explicitly time-dependent

int protons=ncoAddSpecies(NCO_PROTON);
ncoAddRadiationProcess(protons,NCO_INJECTION,NCO_VARIABLE,spec_protons);
ncoAddRadiationProcess(protons,NCO_COOLING,NCO_VARIABLE,loss);
ncoAddRadiationProcess(protons,NCO_ESCAPE,NCO_VARIABLE,escape);
ncoAddInjectionProcess(protons,NCO_SOPHIA_FAST,NCO_VARIABLE,NCO_VARIABLE,protons,photons);
// here photohadronic cooling as discrete energy loss -> re-injection at lower E

int neutrons=ncoAddSpecies(NCO_NEUTRON);
ncoAddInjectionProcess(neutrons,NCO_SOPHIA_FAST,NCO_VARIABLE,NCO_VARIABLE,protons,photons);
ncoAddRadiationProcess(neutrons,NCO_ESCAPE,NCO_VARIABLE,escape);
ncoAddInjectionProcess(neutrons,NCO_SOPHIA_FAST,NCO_VARIABLE,NCO_VARIABLE,neutrons,photons);

// Add photohadronic re-injection of neutrons to protons (only once neutrons are defined)
ncoAddInjectionProcess(protons,NCO_SOPHIA_FAST,NCO_VARIABLE,NCO_VARIABLE,neutrons,photons);
```

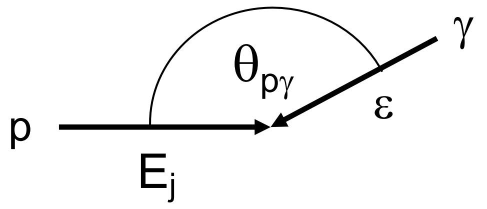
```
// Wrapper for energy loss as a function of log10 x, t; s points to the species
double loss(double x,double t,void* s)
{
  nco_species* spec=(nco_species*)s;
  if(fabs(spec->particlecharge)>0.01)
  {
    double lrate=0.0;
    lrate+=
      ncoComputeSynchrLossRate(spec->particlemass,spec->particlecharge,B, pow(10.0,x));
    lrate+=ncoComputePairProdLossRate(0.938,1.0,spec_photons_wrapper,pow(10.0,x));
    lrate+=1.0/tad; // Adiabatic losses
    return lrate;
  }
  else return 0.0;
}
```

ncoFunctionName:
function from NeuCosmA library

NCO_NAME: constant defined in NeuCosmA library

Efficient treatment of pγ interactions – historical

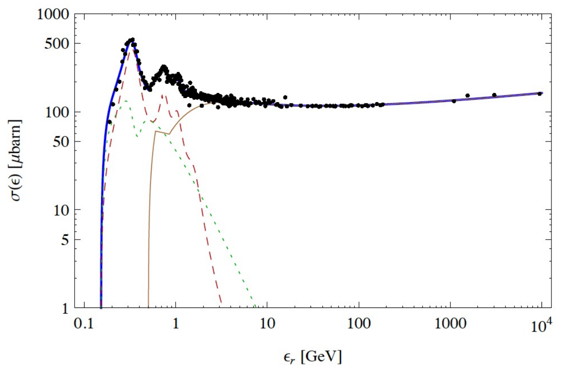
Hümmer et al approach based on SOPHIA – current baseline in AM^3



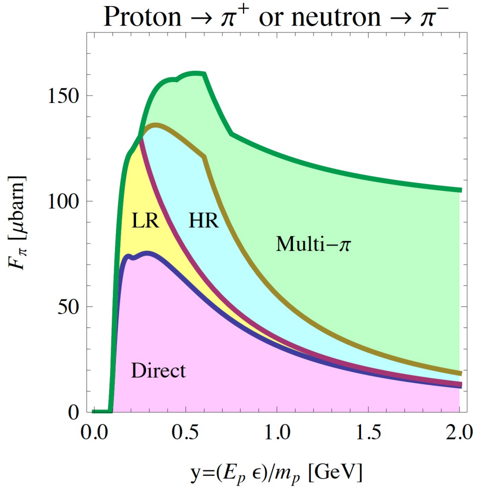
- In general: triple integration needed for re-injection over
  - Pitch angle (typical isotropic distribution), typically integrated out
  - Photon energy
  - Energy other species (re-distribution function!)

$$Q_{ji}(E_i) = \int dE_j\, N_j(E_j)\, \Gamma_j^{\mathrm{IT}}(E_j)\, \frac{dn_{j\to i}^{\mathrm{IT}}}{dE_i}(E_j, E_i)$$

$$\Gamma^{\mathrm{IT}}_{p\gamma\to p' b}(E_p) = \int d\varepsilon \int_{-1}^{+1} \frac{d\cos\theta_{p\gamma}}{2} (1-\cos\theta_{p\gamma})\, n_\gamma(\varepsilon, \cos\theta_{p\gamma})\, \sigma^{\mathrm{IT}}(\epsilon_r)$$



Pitch-angle averaged X-sec



- Often used: Kernels, the spectra can be folded with; **double** integration!
- Idea: Give up the usual Greens-function idea using power law or thermal nature of spectra, discretize the re-distribution function integral in few steps, split processes physics-motivated.
  Single integration, summed over a number of interaction types (IT)

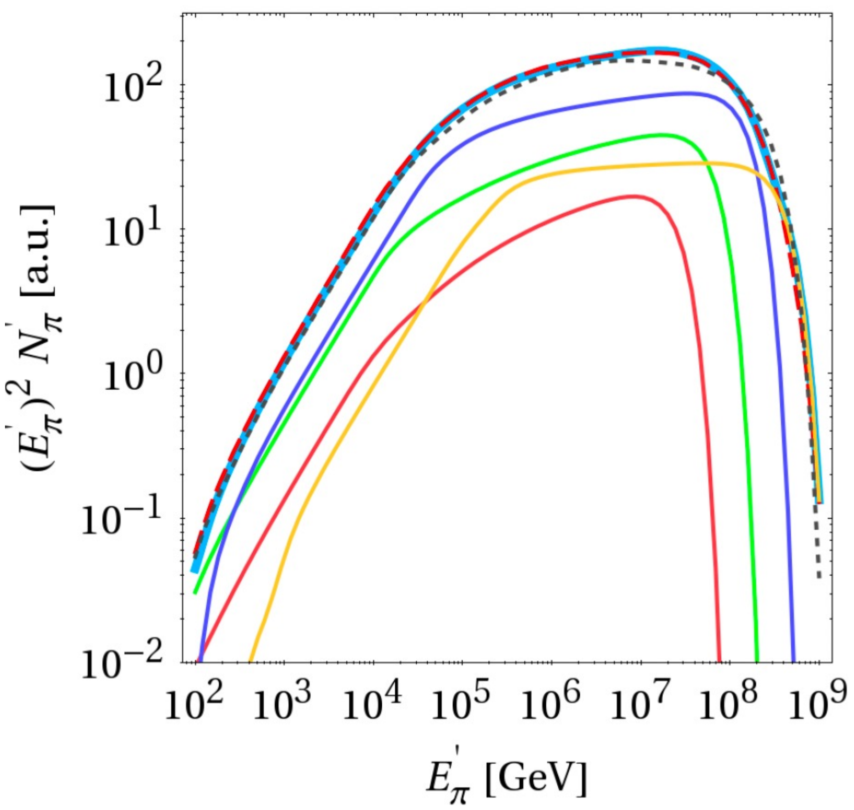
Hümmer, Rüger, Spanier, Winter, Astrophys. J. 721 (2010) 630

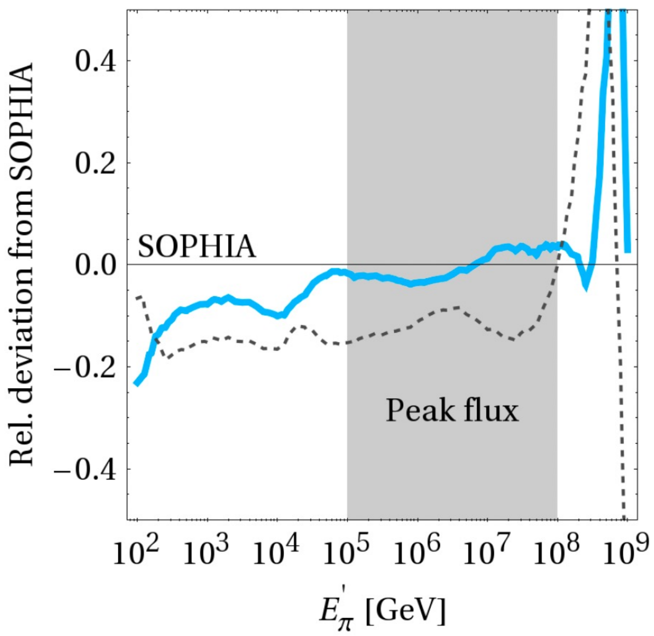
Efficient treatment of photonuclear interactions – NEW

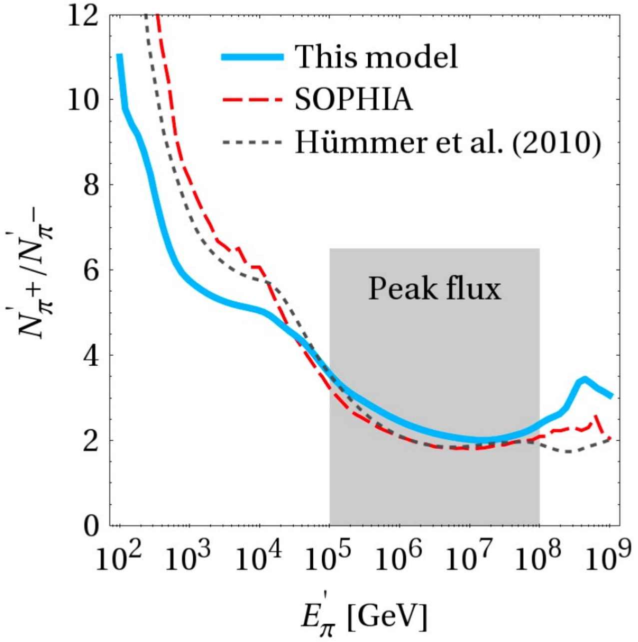
Current baseline in NeuCosmA (but: Hümmer at al method can be also chosen)

- Physics-motivated splitting turned out to be impractical if thousands of nuclear isotopes and processes have to be treated
- New approach: Discretize re-distribution-function-integral in N steps automatically (N can be chosen depending on needed precision; N = 24 can be compared to **Hümmer et al, 2010**)
- Re-applied to pγ interactions and SOPHIA: turned out to be much faster and more precise

N=4







Biehl, Boncioli, Fedynitch, Winter, Astron. Astrophys. 611 (2018) A101, see Appendices

Summary and outlook

NeuCosmA

- Library to set up complicated nuclear systems and compute the related neutrino production
- Solver: Stability, precision and **performance**
- Photon fields external input
- Coupling with AM^3 being pursued in iterative approach

Photohadronic interactions

- Hümmer et al, 2012 approach developed further to extend it to large classes of isotopes, Biehl et al, 2018
- Efficiency and precision high, but comparison to other approaches at kernel level difficult

Ap interactions

- Implementations largely based on Kelner, Aharonian, 2006
- Secondary re-injection may require some thoughts

Discussion

- **FAIR** Data Principles (Findable, Accessible, Interoperable, and Reusable):
  how long can we afford to circumnavigate that?
- Code development typically too much effort for a single PhD student, Postdoc or even group
- “Competition” (source physics) e.g. CRPropa

Outlook/thoughts/wish list

- Open source development platform, e.g. based on some exitsing code like AM^3?
- Separate PDE solver from radiation processes (allows separate approach for different problems and solvers, such as spatially resolved models, acceleration, Monte Carlo)
- Clearly defined interface functions. Switches for different approaches. Modularity.
- Easy identification of logical errors, computational bottlenecks etc

BACKUP

Recap: Interaction rate

- Interaction rate

$$\Gamma^{\mathrm{IT}}_{p\,\gamma\to p'\,b}(E_p) = \int d\varepsilon \int\limits_{-1}^{+1} \frac{d\cos\theta_{p\gamma}}{2}\,(1-\cos\theta_{p\gamma})\,n_\gamma(\varepsilon,\cos\theta_{p\gamma})\,\sigma^{\mathrm{IT}}(\epsilon_r)$$

- For isotropic target photons:

$$\Gamma^{\mathrm{IT}}_j(E_j) = \int d\varepsilon\, n_\gamma(\varepsilon)\, f^{\mathrm{IT}}(y)$$

Constant for photon spectrum constant in time!

Pre-computable function.

$$f^{\mathrm{IT}}(y) \equiv \frac{1}{2y^2}\int\limits_{\epsilon_{\mathrm{th}}}^{2y} d\epsilon_r\,\epsilon_r\,\sigma^{\mathrm{IT}}(\epsilon_r)$$

Integral corresponds to scattering angle integration, y related to “average” CM energy

$$y \equiv (E_j\varepsilon)/m_A$$

The method

- From our assumptions, we obtain ($x=E_i/E_j$)

Integral over σ (universal!)

$$Q_{ji}(E_i) = \int \frac{dx}{x} N_j\left(\frac{E_i}{x}\right) \int d\varepsilon n_\gamma(\varepsilon) f(y) \frac{dn_{j\to i}}{dx}(x,y)$$

- We introduce different ITs with $\tilde{x} = \log_{10} x$ and equally spaced $\tilde{x}$ in reasonable range:

$$Q_{ji}(E_i) = \sum_{\tilde{x}_{\rm IT}} \Delta\tilde{x}\, N_j\left(\frac{E_i}{10^{\tilde{x}_{\rm IT}}}\right) \frac{m_A}{E_i} \times$$

$$\times \boxed{\int dy\, n_\gamma\left(\frac{m_A y 10^{\tilde{x}_{\rm IT}}}{E_i}\right) f(y) \frac{dn_{j\to i}}{d\tilde{x}}(\tilde{x}_{\rm IT}, y)}$$

Integrals can be precomputed if n_γ constant in time

Re-distribution functions (functions of y!)

Biehl, Boncioli, Fedynitch, Winter, Astron. Astrophys. 611 (2018) A101, see Appendices