LISE++ Prefragment search and Universal Parameterization Updates

Excitation-energy control for prefragment search and UP momentum-distribution convolution

Version notes covered: 18.4.4 - 18.4.8, May 20-21, 2026

Source reviewed: o_Fireball.cpp; screenshots supplied from the APF and UP dialogs

What is Universal parameterization: https://lise.frib.msu.edu/doc/Universal_parameterization_Convolution_method_LISE.pdf

Purpose of the update. The APF search and the Universal Parameterization (UP) convolution now use a user-visible excitation-energy prescription. The user can keep the historical fixed slope, E* = slope x dA, or take the mean excitation energy from the current Abrasion-Ablation settings. This makes the selected prefragment, the evaporation competition, and the UP momentum-distribution parameters more physically consistent and easier to test.

1. What changed

The new controls connect two previously semi-hidden choices: the prefragment-search excitation scale and the excitation scale used inside the UP momentum-distribution model. In practice, the same button now appears in the APF/prefragment dialog and in the UP convolution dialog.

Version	Change	User-visible effect
18.4.4	E* buttons added for UP in the APF and UP dialogs	The relevant excitation setting is directly accessible from both workflows
18.4.5	New globals for UnivParameterization -> Excitation Energy	Fixed-slope mode and current-AA mode are stored explicitly
18.4.6	New getExDistrStat(const Celement& CeUse, int method)	A single helper returns excitation statistics and fills FBall->disE
18.4.7	New TApf_ExUP_dlg dialog	The user can choose E* = slope x dA or current AA settings
18.4.8	New Fireball::calcExcitation_Apf(const Celement& Ce)	APF search uses the selected excitation-energy prescription

Recommended short release-note wording: The APF search and UP convolution dialogs now expose a common excitation-energy setting. Users can retain the original 14 MeV per abraded nucleon prescription or use the active Abrasion-Ablation excitation-energy model.

2. Dialog changes and workflow

2.1 APF/prefragment dialog

The APF dialog now shows the selected excitation-energy prescription in the prefragment-search panel. For the historical mode the displayed button reads similar to **Ex = 14.0 * dA**, making clear that the prefragment search is driven by excitation energy per abraded nucleon and that this same value is used by the UP model.

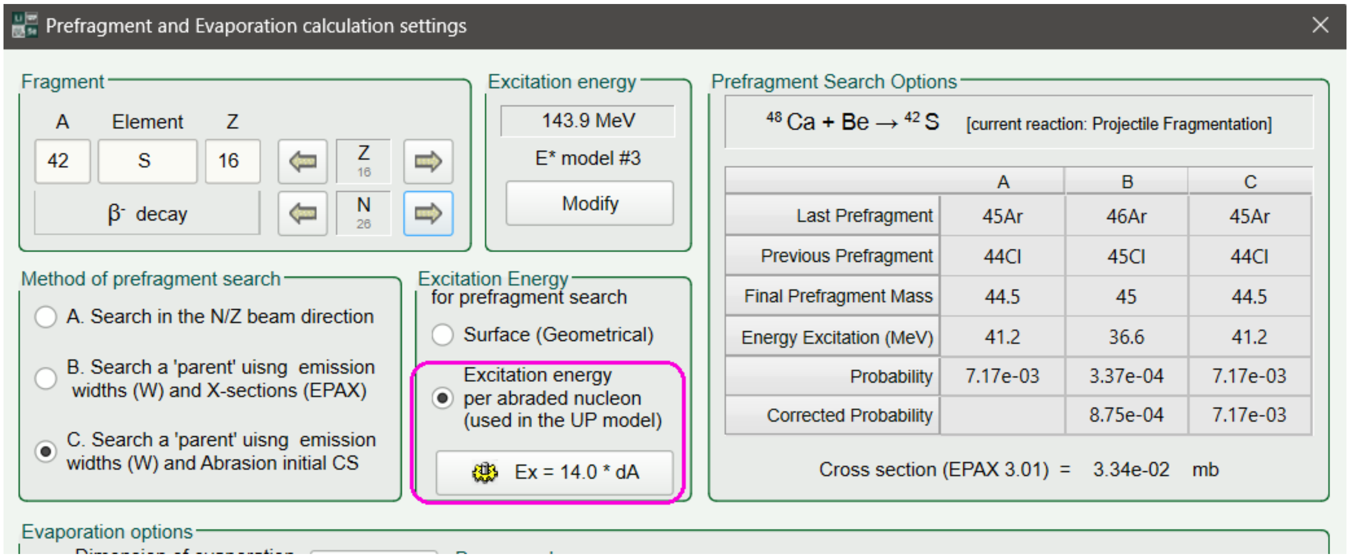

Figure 1. APF/prefragment settings schematic: the new E control is placed next to the prefragment-search method and the A/B/C prefragment comparison table*

The table on the right remains useful as a diagnostic: changing the E* prescription can change the selected last/previous/final prefragment, the emission probabilities, and therefore the effective parent used for the momentum-distribution calculation.

2.2 Excitation-energy setting dialog

The new dialog **TApf_ExUP_dlg** contains the two intended operation modes:

- Manual fixed slope: E* = slope x dA, where dA is the number of abraded nucleons
- Use current Abrasion-Ablation settings: take the excitation-energy statistics from the currently selected AA excitation model

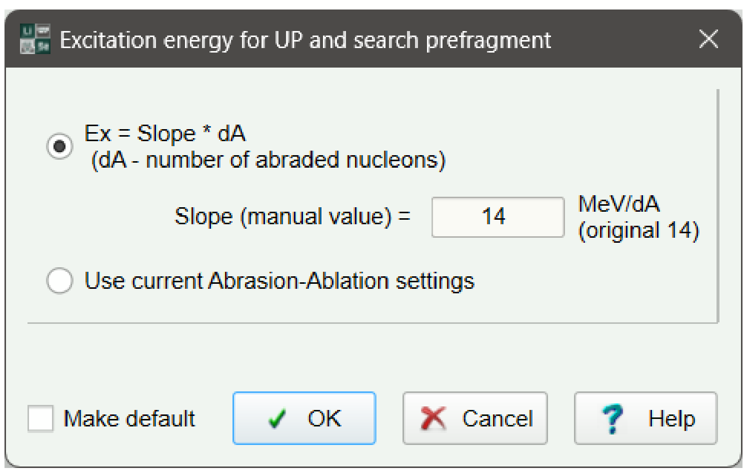

Figure 2. Excitation-energy settings schematic for UP and prefragment search: fixed slope or current Abrasion-Ablation settings

The default manual value of 14 MeV/dA preserves the original UP behavior, so old calculations remain reproducible unless the user intentionally switches to the current AA model.

2.3 UP convolution dialog

The UP convolution dialog now exposes the same excitation control in the row labeled **Excitation from the Abrasion model**. This makes the connection explicit: the excitation choice modifies the separation/excitation scale that enters the convolution parameters.

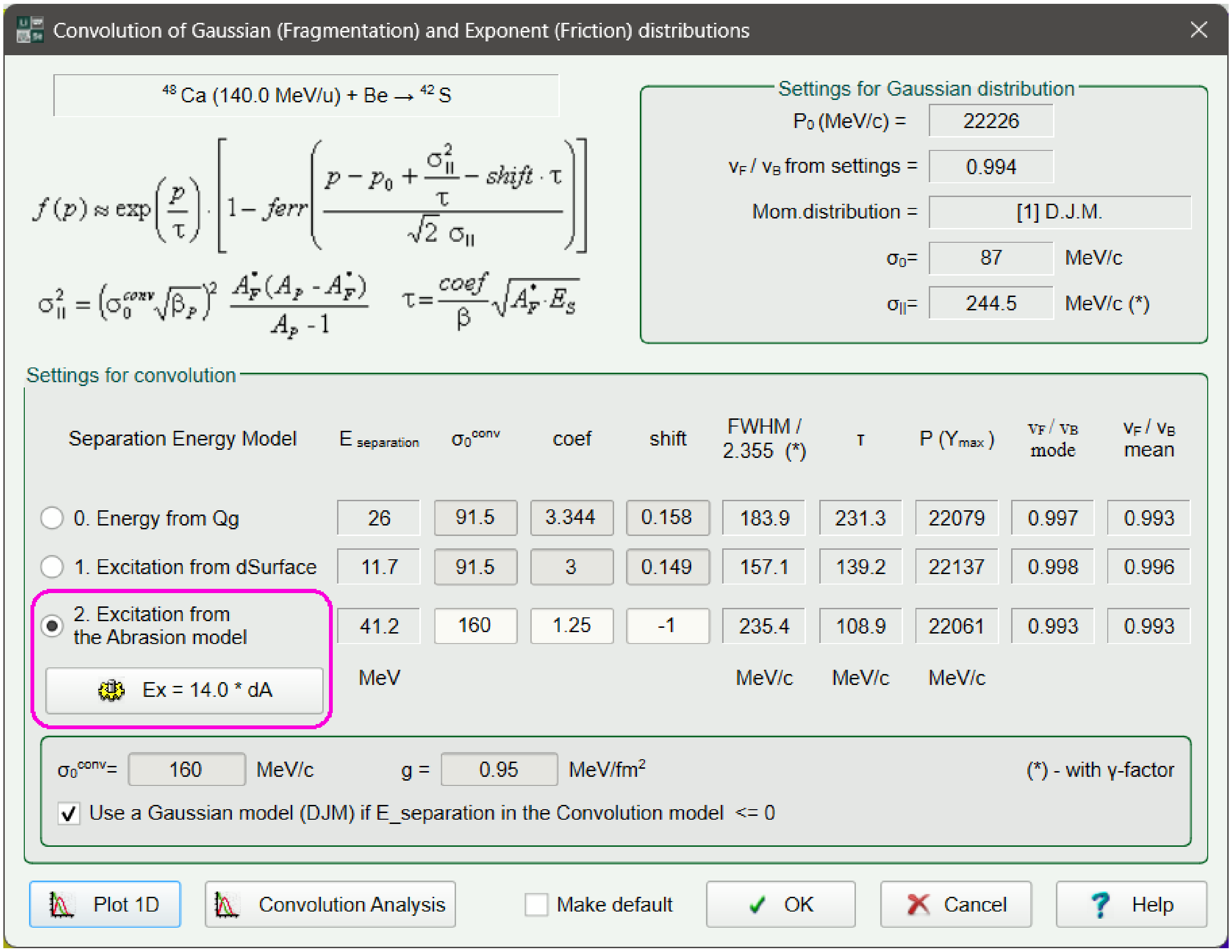

Figure 3. UP convolution dialog: the E control is available directly in the convolution settings*

3. Physics motivation

The change is physically motivated because the APF search is not just a bookkeeping step. The selected abrasion prefragment defines the starting point for evaporation and, in the UP model, influences the longitudinal momentum distribution through the excitation/separation-energy scale.

- Projectile fragmentation first produces an excited abrasion prefragment with reduced mass and charge relative to the projectile
- The excitation energy controls the competition between neutron, proton, and alpha emission widths during the search for the parent/prefragment
- The same excitation scale enters the UP convolution through the friction/exponential component and the effective separation-energy term
- Therefore, different E* prescriptions can move the selected APF and change the predicted momentum centroid, width, and transmitted yield

Why keep the fixed 14 MeV/dA option? It is the historical, stable UP prescription. It is very useful for reproducibility, quick estimates, and comparisons with older LISE++ calculations.

Why add the current-AA option? Modern AA calculations may use EDM/exponential, Gaimard, simplified, Gaussian/surface, or log-normal excitation-energy distributions. If the evaporation calculation is calibrated with one of these models, using the same model in the APF search avoids a hidden inconsistency between the evaporation physics and the momentum-distribution physics.

Practical consequence. For a case such as 48Ca + Be -> 42S, a parent around A = 45 corresponds to dA about 3. With the original slope, the E* scale is about 42 MeV. Changing the E* model can shift the balance of candidate parents in the A/B/C search table and can therefore change the UP momentum distribution used in transmission calculations.

4. Implementation summary from o_Fireball.cpp

4.1 New global-control logic

The new global values separate the mode selection from the numerical value:

```
int    G_ExcitationApf1_type;  // 0 - constant value mode; 1 - current AA
double G_ExcitationApf1_MeV;   // constant value in MeV/dA for type = 0
```

The intended meaning is:

- G_ExcitationApf1_type = 0: use E* = dA x G_ExcitationApf1_MeV
- G_ExcitationApf1_type = 1: use the mean of the current AA excitation-energy distribution

4.2 Fireball::calcExcitation_Apf

The new helper centralizes APF excitation-energy evaluation. In simplified pseudocode:

```
double Fireball::calcExcitation_Apf(const Celement& Ce)
{
    if (G_ExcitationApf == 0)
        return SurfaceEnergyDbl(Ce.A());

    if (G_ExcitationApf1_type == 0) {
        dA = max(abs(beam->A - Ce.A), 1);
        return dA * G_ExcitationApf1_MeV;
    }

    DistrStat stat = getExDistrStat(Ce, G_MethodEE);
    return stat.mean;
}
```

This is the most important code-level change: the APF search no longer needs to know how each excitation-energy model is constructed. It only asks for the excitation scale appropriate to the selected mode.

4.3 getExDistrStat

The helper **getExDistrStat(const Celement& CeUse, int method)** provides a single switch for excitation-energy statistics. It returns a **DistrStat** object and fills the internal distribution used by the Fireball class.

method	Excitation model used
0	Surface / geometrical excitation-energy distribution

1	Gaimard excitation-energy model
2	Simplified excitation-energy model
3 or default	Exponential / EDM excitation-energy model
4	Log-normal excitation-energy model

```
DistrStat Fireball::getExDistrStat(const Celement& CeUse, int method)
{
    switch(method) {
    case 0: return SurfaceEnergy(CeUse.A(), true, true, true);
    case 1: return ExcitationEnergy_Gaimard(&CeUse);
    case 2: return ExcitationEnergy_simplified(&CeUse);
    case 4: return ExcitationEnergy_LogNormal(&CeUse);
    case 3:
    default: return ExcitationEnergy_Exponent(&CeUse);
    }
}
```

4.4 Where the APF search uses the new value

Inside **Fireball::SearchPrefragment**, after each candidate step the code recalculates the excitation energy for the current trial prefragment:

```
Excitation = calcExcitation_Apf(Sapf);
Sapf.Widths(Excitation, SCelement::sc_justNuclei);
...
while (SumExcitation < Excitation && Sapf.A() < beam->A());
```

The loop stops when the accumulated separation, Coulomb, and kinetic-cost estimate reaches the excitation energy supplied by the selected APF/UP mode. Thus, the selected E* prescription can change the interpolation point between the previous and current prefragment and can change the final APF mass used in the calculation.

5. Recommended user-facing description

The following paragraph can be used in the LISE++ help or website documentation:

Excitation energy for the APF search and Universal Parameterization. For Universal Parameterization calculations, LISE++ estimates the abrasion prefragment before evaporation and uses this prefragment information in the momentum-distribution convolution. The excitation energy used in this search can be set either to the original fixed form, E* = slope x dA with slope = 14 MeV/dA by default, or to the mean excitation energy from the active Abrasion-Ablation model. The fixed-slope option preserves historical UP calculations, while the current-AA option makes the APF search consistent with the selected evaporation/excitation-energy model.

6. Suggested validation tests

1. Run a reference UP calculation with fixed slope = 14 MeV/dA and verify that older results are reproduced within numerical precision
2. Switch to current AA settings and check that the APF table changes only when the selected AA excitation model predicts a different mean E* for the trial prefragment
3. For one light and one heavy projectile-fragmentation case, compare the UP momentum centroid, FWHM, and transmission before and after changing the E* mode
4. In MC mode, verify separately the central-target UP option and the event-by-event reaction-position option, because this update changes the UP distribution itself but not the MC depth-sampling logic
5. Save one LISE++ file with manual slope and one with current-AA mode to confirm that G_ExcitationApf1_type and G_ExcitationApf1_MeV are persisted correctly

7. Developer note from source review

In the reviewed source, the final APF-loop stopping condition uses **calcExcitation_Apf(Sapf)**. One candidate-parent branch still contains a local legacy excitation estimate for trial widths:

```
if(G_ExcitationApf == 0 )
  Excitation = SurfaceEnergyDbl(Ce.A());
else {
  if(G_ExcitationApf1_type==0)
    {
    double dA=qFabs( beam->Mnucl() - Ce.A());
    if(dA<1) dA=1;
    Excitation = dA * G_ExcitationApf1_MeV;
    }
  else {
      DistrStat stat = getExDistrStat(Ce, G_MethodEE);
      Excitation = stat.mean;
    }
  }
```

If the intended behavior is full consistency of temporary trial-parent weights with the new dialog choice, this local estimate should probably be routed through **calcExcitation_Apf(*Cpn[i])** as well. If the old estimate is intentionally retained for ranking stability, it is worth adding a short code comment so the difference is clear later.