M1. Diffusion solver module

M1.1 Slot, options and selection

This module fills the diffusion-solver slot of the spine's module-slot table (spine 7.14). It specifies the options, how one is selected and validated at startup, the timestep pair the selected option supplies to the rest of the model, and the algorithm of the selected option.

Block	Activity	Interval
Startup, after construction or load and before initialization	Solver construction, timestep assignment, synchronization and accuracy validation (M1.1, M1.2)	Once per run and on every load
Diffusion loop, between the two split half-steps	Field advance: transforms, derived assignment of CCL5 and CXCL9, post-solve binding sweep, grid exchange (M1.3)	Every Δt_d; N_d times per cellular timestep
End of each split half-step	Smoother (M1.3, The smoother)	Twice per Δt_d, spectral options only

The solver is invoked by no agent and reads no agent state. It reads no caseation state, cell density or compartment counter. Its inputs are the parameters, the grid dimensions, the module gating flags and the two grid objects.

Reads:

- Current grid: TNF, CCL2, IL-10, TGFBActive, TGFBInactive; shedTNFR2 under M3; Etanercept, Infliximab and the four complex species under M7; the antibiotic free and bound grids under M6.
- The four host diffusivities and five degradation constants (M1.6, M4.9).
- `Core_Mac_dCCL2`, `Core_Mac_dCCL5` and `Core_Mac_dCXCL9`, as two quotients only (M1.3, Derived assignment).
- The `Diffusion_FFT_*` timestep and smoother parameters, and the timestep pair of the explicit option (M1.1).

Writes:

- Next grid: the same fields as read.
- Next grid: CCL5 and CXCL9, by pointwise assignment from CCL2 (M1.3, Derived assignment).
- The current-versus-next grid pair, exchanged once per field advance (M1.3, Non-transformed fields and grid buffering).
- The diffusion and molecular timesteps and their step counts, at startup and on load only (M1.1).

The option is selected by the command-line option `--diffusion`, whose value names a member of the solver enumeration.

Value	Option	Specified in
4	Alternating-direction explicit	M1.4
5	Spectral (FFT), default	M1.3
6	Spectral (FFT), estimate planning	M1.3
7	Crank-Nicolson alternating-direction implicit; two-dimensional build only	M1.5

Selection

The selection routine acts in sequence:

1. If a solver object exists and its method differs from the requested one, the object is destroyed.
2. Δt_d and Δt_m are assigned from the parameter group of the requested method.
3. If no solver object exists, one of the requested type is constructed.

The step counts are then derived by integer division:

$$N_m = \left\lfloor \frac{\Delta t_d}{\Delta t_m} \right\rfloor, \qquad N_d = \left\lfloor \frac{\texttt{AGENT_TIME_STEP}}{\Delta t_d} \right\rfloor.$$

With `AGENT_TIME_STEP` = 600 s and the spectral pair of 60 s and 6 s, N_m = 10 and N_d = 10. The program enforces divisibility (M1.2), so the division leaves no remainder and the interval advanced per cellular timestep equals `AGENT_TIME_STEP`.

Selection runs after the simulation object is constructed or a saved state is loaded, after command-line options are processed, and before initialization. The order is required because the validation of M1.2 and the computed-parameter derivations (spine 7.0, Computed parameters) consume the timesteps, and initialization consumes computed parameters.

The placeholder and saved states

The simulation constructor installs a value 4 solver before any selection, because loading a saved state needs a solver object to exist. Selection replaces it unless value 4 is requested. Because the placeholder is installed through the selection routine, the explicit option's timestep pair is read, and its 60 s ceiling enforced (M1.4), at construction on every run and every load. A parameter file with `Diffusion_ADE_timestepDiffusion` above 60 s therefore fails at construction even when the spectral option is requested. A reimplementation may omit this check without changing any trajectory.

Only the method enumeration value is saved with a simulation state. On load it is passed to the selection routine, which replaces the placeholder if it differs. Without it, a loaded state would run under the constructor's solver (M1.1, Saved state).

The Crank-Nicolson option is compiled only in the two-dimensional build; the other options exist in both builds (M5).

Saved state

The solver object is not saved with a simulation state; only the method enumeration value is (M1.1, The placeholder and saved states). The timesteps and step counts are saved as simulation members, then overwritten from the parameters by the selection routine on load. The reloaded timesteps equal the saved ones only if the parameter file is unchanged between save and load: a changed file changes the timesteps on reload.

The solver holds no state that evolves during a run. Its plans, buffers and propagators depend only on the parameters and the grid dimensions. A reimplementation must keep the solver a pure function of the parameters, the grid dimensions and the two grid objects; a solver that adapted its timestep or cached a previous solution would have to be saved.

M1.2 Timesteps and startup validation

Each option declares its own timestep pair. All three pairs, and both smoother parameters, are required in the parameter file and loaded at startup, whichever option is selected. The selected option's pair sets the timesteps. The explicit option's pair is also used by the selection routine at construction on every run and every load, which enforces its 60 s ceiling (M1.1); its molecular timestep is not checked. The Crank-Nicolson pair is used only if that option is selected.

Option	Δt_d	Δt_m	**N_d**	**N_m**
Spectral, both values	60 s	6 s	10	10
Alternating-direction explicit	30 s	5 s	20	6
Crank-Nicolson ADI	60 s	3 s	10	20

The timesteps are properties of the numerical method, not of the biology. No biological rate is expressed in units that depend on them; secretion rates are per second and are multiplied by the half-step at the point of use. The pair is nonetheless load-bearing. Every secretion, consumption and receptor-sink term in S6 and S7 is multiplied by a quantity derived from it, and Δt_m sets how often the sinks act. The options are therefore not interchangeable when results are compared.

Synchronization conditions

Startup validation tests these conditions and reports any failures together as one fatal error:

- `SECONDS_PER_DAY` is divisible by `TIME_STEPS_PER_DAY` (86400/144 = 600 s). Both are compile-time constants, so this guards against an inconsistent recompilation, not a parameter choice.
- Δt_d divides the cellular timestep exactly (600/60 = 10).
- Δt_m divides the cellular timestep exactly (600/6 = 100).
- Δt_m divides Δt_d exactly (60/6 = 10).
- N_m is even, so that both halves of the split step hold the same number of molecular substeps. This parity test is evaluated only once the Δt_m-divides-Δt_d test has passed.

All three declared pairs satisfy every condition.

Molecular timestep ceiling

Δt_m must not exceed `MOLECULAR_ACCURACY` = 30 s, and must not exceed Δt_d. The constant is the accuracy limit for the per-agent ordinary differential equations. It constrains a module that is disabled in the default configuration, but it is enforced unconditionally. All three pairs satisfy both. The coarse IL-10 sink becomes unstable for Δt_m above about 29.7 s at the sampled extreme, which is below the 30 s ceiling (S6.8).

Accuracy heuristic

A further test compares the diffusion timestep against a heuristic limit:

$$\Delta t_d < \frac{\kappa}{D_{\mathrm{TNF}}\,\ell}, \qquad \ell = \frac{2}{\Delta x^2} = 5 \times 10^5 \ \mathrm{cm}^{-2},$$

where κ is a per-option factor and ℓ a fixed geometric quantity from the grid specification. Exceeding the limit terminates startup with a fatal error.

Option	κ	Limit	Declared Δt_d
Alternating-direction explicit	2.0	76.9 s	30 s
Spectral, both values	10^4	3.846×10^5 s	60 s
Crank-Nicolson ADI	10^4	3.846×10^5 s	60 s

The constraint is slack by nearly four orders of magnitude under the spectral option and by a factor of about two and a half under the explicit option.

The check reads `Core_diffusivityTNF` alone.

Drug limits

Under M6 the validation also applies two drug limits: the accuracy heuristic with the largest drug diffusivity in place of D_TNF, and a permeability ratio

$$\rho = \mathrm{perm}_{\max} \cdot A_{\mathrm{cmpt}} \cdot \frac{\Delta t_d}{2} \cdot \frac{1}{1000\,V},$$

with the half-step because drug permeability is applied once per half-step. These validation checks reject $\rho \geq 1$, and run only on a load or restart, against the saved values. In a new run each drug's permeability is instead tested when the drug is created, rejecting $\rho > 1$, and again on a regular reload (M6.4).

M1.3 Spectral option (values 5 and 6)

Each field is advanced in one step per Δt_d by a sine or cosine transform, with decay folded into the propagator. It supplies $\Delta t_d = 60$ s and $\Delta t_m = 6$ s. Value 5 is the spectral (FFT) option and the default; value 6 is the spectral option with estimate planning, which gives identical transforms in every run (M1.3, Planning strategy).

Value 6 is not a distinct scheme. It shares every line of its implementation with value 5 and differs only in the planning flag passed to the transform library (M1.3, Planning strategy). This section therefore applies to both, which are called jointly the spectral option.

Fields presented to the solver

The solver receives the whole grid and decides per field what to do with it.

Field	Treatment	Boundary	Gate
TNF	Transformed	Zero concentration	None
CCL2	Transformed	Zero concentration	None
IL-10	Transformed	Zero concentration	None
TGFBActive	Transformed	Zero concentration	None
CCL5, CXCL9	Assigned pointwise from CCL2 (M1.3, Derived assignment)	n/a	None
TGFBInactive	Copied forward, then decayed (M1.3, Non-transformed fields and grid buffering)	n/a	None
shedTNFR2	Transformed	Zero concentration	TNF receptor flag, M3
Etanercept, Infliximab	Transformed	No flux	Immunomodulation gate, M7
EtanTNF, Ifx1TNF, Ifx2TNF, Ifx3TNF	Transformed	No flux	Immunomodulation gate, M7
Antibiotic free-drug grids	Transformed	No flux	Antibiotic gate, M6
Antibiotic caseum-bound and epithelium-bound grids	Decayed	n/a	Antibiotic gate, M6
O_2	Not handled	n/a	n/a
Collagen	Not handled	n/a	None

The immunomodulation gate is the M7 module flag conjoined with $t \geq$ `Mod_dosageStartTime`, so the six anti-TNF species are not transformed before dosing begins even when the module is enabled.

Four live transforms

Four fields are transformed per diffusion step by default: TNF, CCL2, IL-10 and TGFBActive. The TGF-β module flag is not passed to the solver, so the TGFBActive transform is ungated. M4 is disabled and nothing secretes into that field, so it is zero throughout the run. Its transform still costs a forward transform, a propagator multiplication and an inverse transform on every diffusion step: ten per cellular timestep, and about 2.2×10^5 over a run of 22176

timesteps.

For a non-zero diffusivity the effect on the model is nil: the transform of zero is zero, and the cutoff (M1.3, Normalisation and the cutoff) keeps it zero. It is one of the four transforms per diffusion step. A reimplementation without TGF-β may omit the transform without changing any trajectory, since the transform uses no random draws and the solver holds no state that depends on it.

Fields the solver does not touch

Oxygen and collagen are neither transformed, nor copied, nor decayed by the solver. Oxygen is written by the consumption and vascular flux terms of S7 in both half-steps around the solve; collagen by the myofibroblast secretion of M4, disabled by default. Their writes are split between the two grid objects, and only one object's history survives (M1.3, Non-transformed fields and grid buffering).

No solver transports oxygen in any configuration: the oxygen diffusion path is compiled out in every solver.

The scheme

Each transformed field C is advanced with the working buffer B of its boundary type and its propagator P:

```
for each compartment p_k of the domain, in row-major order:
    assemble:     B[k] <- C_current(p_k)

forward:          B <- F(B)                          in place
propagate:        B[k] <- B[k] * P[k] * s            for every k;  s = 1/(4N)
inverse:          B <- F^-1(B)                        in place

for each compartment p_k:
    disassemble:  C_next(p_k) <- 0 if B[k] <= cutoff, otherwise B[k]
                  B[k] <- 0
```

The buffer holds exactly N = n_x n_y entries. The halo is excluded from the transform, and the boundary condition comes from the choice of transform (M1.3, Boundary conditions), not from the halo. Disassembly applies the cutoff (M1.3, Normalisation and the cutoff) and zeroes each buffer entry as it is consumed.

Fields do not couple within a solve, so the order in which they are advanced is immaterial with two exceptions: the derived assignment (M1.3, Derived assignment) must follow the disassembly of CCL2, and the binding sweep (M1.3, Post-solve binding sweep) must follow the transform of every species it touches. A reimplementation may otherwise reorder or parallelise the per-field work.

Timestep tolerance

Diffusion and first-order decay are applied together as one exponential per transform mode over Δt_d (M1.3, The propagator), so the field update has no stability restriction on Δt_d. That exponential is exact for the diagonal operator the propagator encodes (M1.3, The propagator).

What limits Δt_d is the resolution of the source and sink terms. They are applied as two half-step impulses per diffusion step, so a larger Δt_d means fewer, larger impulses. The method stays accurate up to a diffusion timestep of half the cellular timestep, 300 s; the value in use is a fifth of that.

The propagator and incorporation of decay

One propagator is built per field from its diffusivity D, its decay constant k_deg, the compartment length Δx, the grid dimensions and Δt_d. There are two constructions, one per boundary type.

For the zero-concentration boundary, the eigenvalue at index (i, j) is

$$P_{ij} = \frac{2D}{\Delta x^2}\left[2 + \frac{k_{\deg}\Delta x^2}{2D} - \cos\left(\frac{\pi(j+1)}{n_y+1}\right) - \cos\left(\frac{\pi(i+1)}{n_x+1}\right)\right],$$

and for the no-flux boundary

$$P_{ij} = \frac{2D}{\Delta x^2}\left[2 + \frac{k_{\deg}\Delta x^2}{2D} - \cos\left(\frac{\pi j}{n_y-1}\right) - \cos\left(\frac{\pi i}{n_x-1}\right)\right].$$

The stored entry is $\hat{P}_{ij} = \exp(-P_{ij}\,\Delta t_d)$, the factor applied in the propagate step, not the eigenvalue itself. Δx is `MICROCMPT_LENGTH_CENTIMETERS` = 20×10^{-4} cm.

These cosine arguments are the eigenvalues of the type-I sine and cosine transforms; the transforms applied are of types II and III (M1.3, Boundary conditions), in both the two- and three-dimensional builds.

Decay folded into the propagator

The term $k_{\deg}\Delta x^2/(2D)$ carries decay. Multiplying through,

$$P_{ij} = \underbrace{\frac{2D}{\Delta x^2}[2-\cos(\cdot)-\cos(\cdot)]}_{\text{diffusion}} + \underbrace{k_{\deg}}_{\text{decay}},$$

so one exponential advances diffusion and first-order decay together over Δt_d, with no splitting between them.

No separate decay sweep exists for any transformed field. The bound-drug decay helper acts only on the antibiotic bound grids, and the copy-and-decay step only on TGFBInactive (M1.3, Non-transformed fields and grid buffering). This is the substantive numerical difference from the explicit option, which decays as a separate multiplicative step after its sweeps (M1.4). A reimplementation that applied decay separately would add a splitting error this scheme lacks. It is small but non-zero, so trajectories would diverge: $k_{\deg}\Delta t_d$ is 0.0948 for TNF and CCL2 and 0.0291 for IL-10, retaining 0.9096 and 0.9713 of each field per diffusion step.

Constants by field

Field	Diffusivity	Decay constant
TNF	`Core_diffusivityTNF`	`Core_kDeg`
CCL2	`Core_diffusivityChemokines`	`Core_ChemokinekDeg`
IL-10	`Core_diffusivityIL10`	`Core_Ikdeg`
TGFBActive	`Core_diffusivityTGFBActive`	`Core_degRateTGFBActive`
TGFBInactive	`Core_diffusivityTGFBActive`	`Core_degRateTGFBInactive`
shedTNFR2	`Molecular_TNF_diffusivityShedTNFR2`	0, a literal

The shedTNFR2 decay constant is zero because that species loses mass by dissociation in the binding sweep (M1.3, Post-solve binding sweep), which returns the TNF; treating the loss as decay as well would destroy mass the model conserves.

The TGFBInactive propagator is built, from the active diffusivity and the inactive decay constant, and never applied: the field is copied, not transformed (M1.3, Non-transformed fields and grid buffering). No `Core_diffusivityTGFBInactive` parameter exists. CCL5 and CXCL9 have no propagators (M1.3, Derived assignment).

Construction

All propagators are built in one pass at the end of the constructor and of the copy constructor, and are never rebuilt. Every input, including the sampled IL-10 diffusivity, is fixed for a run. A diffusivity that varied with local cell density, as the `minDiffusivity` parameters anticipate (M7.6), could not use precomputed propagators; the drug fields cannot change diffusivity during a run. Antibiotic propagators are the exception to the one-pass construction: they are built in each antibiotic object's own constructor (M6.5).

The propagators for shedTNFR2 and the six anti-TNF species are built at construction whether or not M3 or M7 is enabled. This is a startup and memory cost only; a reimplementation may build them lazily or conditionally.

Storage stride

The zero-concentration construction stores entry (i, j) at $i \cdot n_y + j$ and the no-flux construction at $i \cdot n_x + j$; the two coincide on a square grid, such as the default 300×300.

Zero diffusivity

A zero diffusivity makes every propagation coefficient not-a-number, under both boundary constructions and in both builds: the prefactor is zero while $k_{\mathrm{deg}} \Delta x^2 / (2D)$ is infinite, or zero over zero when k_deg is also zero. A not-a-number value fails the cutoff test and is written to the next grid. The active TGF-β field is transformed in every configuration, so a zero `Core_diffusivityTGFBActive` makes that field not-a-number, as a zero diffusivity does a transformed drug field (M7.12). No diffusivity in the default values is zero.

Boundary conditions

GranSim has no per-field boundary setting. The boundary condition follows from the transform pair a field is passed through, so the field table (M1.3, Fields presented) is the whole specification.

Pair	Forward	Inverse	Basis	Boundary realised
Zero concentration	`FFTW_RODFT10`	`FFTW_RODFT01`	Sine	C = 0 at the edge (Dirichlet)
No flux	`FFTW_REDFT10`	`FFTW_REDFT01`	Cosine	$\partial C / \partial n = 0$ at the edge (Neumann)

Each pair uses the same transform kind in every dimension, so the condition is uniform around the domain; different edges cannot take different conditions. Every sine basis element vanishes at the edge and every cosine basis element has zero normal derivative there, so no padding of the grid is needed for either condition.

The spectral solver reads and writes fields through the padded accessor but never reads a halo entry; the halo belongs to the grid (spine 2.2).

Consequence in the default configuration

All four transformed fields take the zero-concentration pair, so TNF, CCL2, IL-10 and TGFBActive lose mass across the domain edge. The derived chemokines inherit this. Oxygen has no boundary condition, having no spatial operator.

Agents and fields treat the domain boundary differently (spine 1.3, 2.2): agents treat the domain as a torus, fields treat its edge as an absorbing wall. The transform library offers a periodic condition, but no option selects it, so no configuration gives fields and agents one convention.

The edge acts as a sink proportional to the edge gradient. For a lesion forming centrally in a 300×300 domain it is negligible; it grows as a lesion nears the edge. The loss is confined to a margin of about three compartments set by the decay length (S6.15).

Normalisation and the cutoff

The transform library computes unnormalised transforms. Each one-dimensional round trip of the chosen pairs multiplies by 2n, giving 4 n_x n_y in two dimensions, the same for both boundary types. The correction is one scalar applied in the propagate step:

$$s = \frac{1}{4N}, \qquad N = n_x n_y,$$

which cancels the round trip exactly. It is a property of the transform, not a model parameter: a reimplementation using another library's conventions, or working in three dimensions (M5), must recompute it.

The cutoff

At disassembly, with b the buffer entry and ϵ = `DIFFUSION_CUTOFF` = 10^{-12},

$$C_{\text{next}}(p) = \begin{cases} 0 & b \le \epsilon \\ b & b > \epsilon \end{cases}$$

The test is one-sided and inclusive. Its declared purpose is to treat negligible values as absent and to keep values from going negative. A spectral round trip can produce small negative values from non-negative input, and the one-sided test clamps them to zero as well; a negative concentration in an S6 occupancy expression would be meaningless. The cutoff is the only guard against negative field values, and it makes the exact-zero branch of the S6.11 TNF logistic reachable for IL-10.

The same cutoff is applied by the bound-drug decay helper and by the explicit option within each sweep. The TGFBInactive copy-and-decay step applies none, so TGFBInactive can hold values below ϵ; decay cannot make a non-negative value negative.

The cutoff is in molecules per compartment, twelve orders of magnitude below one molecule. It is a numerical floor, not a biological threshold, and no rule has a threshold near it.

Derived assignment of CCL5 and CXCL9

Immediately after CCL2 is disassembled into the next grid, and before IL-10 is transformed, every compartment's CCL5 and CXCL9 are overwritten:

$$C_{\text{CCL5}}(p) \leftarrow r_5\, C_{\text{CCL2}}(p), \qquad C_{\text{CXCL9}}(p) \leftarrow r_9\, C_{\text{CCL2}}(p), \qquad r_5 = \frac{\texttt{Core_Mac_dCCL5}}{\texttt{Core_Mac_dCCL2}}, \quad r_9 = \frac{\texttt{Core_Mac_dCXCL9}}{\texttt{Core_Mac_dCCL2}}.$$

By default $r_5 = 1.0$ and $r_9 = 2.0$ (S6.14). The ratios are recomputed on every call, not cached. The assignment is unconditional and ungated, and it is an assignment, not an increment.

Direct writes to CCL5 and CXCL9 are overwritten at the next solve, except those of the final secretion half-step, which agents read (S6.14). The writers are the macrophage chemokine helper, caseum secretion (S5.6) and, under M3 NF-κB dynamics, the molecular-substep write-back, which uses a literal 1 : 1 : 2; the solve's ratios are recomputed from the macrophage secretion rates on every call. This holds under all three solvers, since it is a property of the diffusion loop rather than of the scheme.

Parameter-file initialisation of these fields is discarded at the first solve (spine Element 5).

The assignment is exact while the S6.14 conditions hold; a reimplementation must perform it inside the loop, as specified here.

The assignment saves two of the six transforms per diffusion step, counting the TGFBActive transform. The relation, its exactness conditions and its consequences for chemotaxis and recruitment are stated at S6.14 and S1.3. The smoother repeats the assignment after each CCL2 relaxation (M1.3, The smoother), and the explicit option fuses it into its CCL2 averaging step (M1.4).

Non-transformed fields and grid buffering

TGFBInactive

The first operation of each field advance copies the whole TGFBInactive storage from the current grid to the next, then multiplies the copy pointwise by the decay factor:

$$C_{\text{next}}(p) \leftarrow C_{\text{current}}(p) \cdot \exp(-k_{\text{deg}}\,\Delta t_d)\,, \qquad k_{\text{deg}} = \texttt{Core_degRateTGFBInactive}.$$

The solver's contract with the timestep

The solver reads the current grid, writes the next grid, then exchanges the two. Writes made during a split half-step go to the current grid. If the solver transforms or copies a field, those writes are carried into the next grid. If it does neither, the field's entry in the next grid keeps whatever it held when that grid object was last current, and the exchange makes it current again. Writes to such a field accumulate separately in the two grid objects, and only one of the two histories survives.

Oxygen is in this position: only the history of the grid object current at the end of the loop survives, one application of each oxygen routine per diffusion step (S7.1, S7.6). The same holds at every exchange the smoother makes (M1.3, The smoother). Collagen is in the same position, written only by M4 and zero by default (M4.4). Under M7 with its transport gate closed, the drug fields are in the same position (M7.6).

No other field written during a half-step is affected: the cytokines are transformed, TGFBInactive is copied, and CCL5 and CXCL9 are overwritten.

Exchange and shared state

The exchange runs once per field advance, N_d times per cellular timestep. Because N_d = 10 is even, a reference to the current grid taken before the diffusion loop still refers to the current grid after it (spine 2.2).

The per-compartment bacterial containers are shared by pointer between the two grid objects and are unaffected by the exchange (spine 2.2). Antibiotic grid objects are likewise shared, and each runs its own exchange at the end of its drug's transform (M6.5).

Post-solve binding sweep

After every transform and before the exchange, one sweep over the domain applies the gated binding blocks to the next grid. The sweep reads and writes the next grid throughout, so within a compartment each block sees the results of the blocks run before it, in the order Etanercept, Infliximab, shed TNFR2. Rate terms are multiplied by the full Δt_d, since the sweep runs once per diffusion step. Every gate is closed by default, so the sweep visits all 90000 compartments and changes nothing. It is live code with nil effect, not unreachable code: each gate is a parameter or a module flag.

- **Etanercept**, gated on `Mod_Etanercept_dosage` $\neq 0$, and **Infliximab**, gated on `Mod_Infliximab_dosage` $\neq 0$. The binding equations, and the sign of the Etanercept TNF update, which differs between solvers, are specified at M7.7. These blocks are gated on the dosages alone, while the anti-TNF transforms are gated on the immunomodulation gate, so binding can run without transport (M7.2).
- **Shed TNFR2 dissociation**, gated on the TNF receptor flag (M3):

$$C_{\text{shed}} \mathrel{-}= C_{\text{shed}}\, K_{D2}\, k_{\text{on2}}\, \Delta t_d, \qquad C_{\text{TNF}} \mathrel{+}= C_{\text{shed}}\, K_{D2}\, k_{\text{on2}}\, \Delta t_d,$$

where K_D2 = `Molecular_TNF_KD2` and k_on2 = `Molecular_TNF_kOn2`, so K_D2 k_on2 is the dissociation rate constant. The decrement is applied first, so the TNF update reads the already-reduced shed value, and the TNF gained is the shed complex lost multiplied by $1 - K_{D2} k_{\text{on2}} \Delta t_d$. M3.3 specifies the shed TNFR2 field.

The smoother

The smoother is an explicit relaxation pass run at the end of each split half-step, outside the field advance, and only under the spectral options; the explicit option declares an empty smoother. It is entered twenty times per cellular timestep, once at the end of each split half-step. At zero steps each call returns after evaluating the immunomodulation gate and the step count.

Each iteration applies a forward-Euler five-point step, with no decay:

$$C_{\text{next}}(i,j) = C(i,j) + \text{Fo}\,[C(i{-}1,j) + C(i{+}1,j) + C(i,j{-}1) + C(i,j{+}1) - 4\,C(i,j)]\,, \qquad \text{Fo} = \frac{D\,\Delta t_s}{\Delta x^2},$$

with Δt_s = `Diffusion_FFT_timestepDiffusionSmoother` and Δx^2 = `MICROCMPT_FACE_AREA_CENTIMETERS_SQUARED` = 4×10^{-6} cm^2. Each field uses the variant matching its transform boundary. The zero-concentration variant reads the halo directly; an uninitialised halo holds zero, which gives an absorbing boundary. The no-flux variant substitutes the centre value for any out-of-range neighbour. The grids are exchanged at the end of each iteration. Diffusivities are read from the parameters, independently of the propagators.

With a non-zero step count the fields relaxed would be:

- TNF, CCL2, IL-10 and TGFBActive, unconditionally, with the derived assignment (M1.3, Derived assignment) after each CCL2 relaxation, so once per iteration per half-step as well as once per solve;
- shedTNFR2 under M3;
- the anti-TNF species under M7;
- the antibiotics under M6, with an extra call to carry forward their non-diffusing grids.

Oxygen is not relaxed, its path being compiled out (M1.3, Fields presented).

By default `Diffusion_FFT_numSmootherSteps` = 0, so no field is relaxed, the smoother timestep is never consulted, the derived assignment inside the loop is never performed and no grid is exchanged.

The explicit step is stable for $\mathrm{Fo} \leq 1/4$ in two dimensions. With Δt_s = 1.2 s, Fo is 0.0156 for TNF and at most about 0.018 for IL-10, both stable. The bound is not checked at startup: the accuracy heuristic (M1.2) examines Δt_d, not Δt_s. M7.6 covers the drug fields.

Planning strategy and determinism

Value	Option
5	Spectral (FFT), default
6	Spectral (FFT), estimate planning; identical transforms in every run

The planning flag is applied to every transform plan.

The value 5 plan is faster and is meant for production, but is not guaranteed to give identical results across runs with the same executable, parameter file, command line and seed. The value 6 plan is slower and is meant for testing; it gives identical results across such runs, and across code versions whose changes are not meant to alter results.

The mechanism is that run-time measurement selects among algorithms that round differently. A difference in the last bits of a field value can flip a threshold comparison when the value sits near the threshold, and the trajectory then diverges. The seven-significant-figure truncation of those comparisons (spine 2.1) absorbs only differences below the seventh figure, and a measurement-dependent algorithm selection gives no bound on how far a difference can grow once fed back through the nonlinear agent rules.

Another transform library, or the same library on other hardware, will not reproduce trajectories bitwise; the guarantee is per executable and machine. A reimplementation must ensure that the transform does not vary between runs of one build. Any strategy that is deterministic given the problem size satisfies this; a strategy that measures and selects does not.

M1.4 Alternative option: alternating-direction explicit

Each field is advanced by two explicit directional sweeps over separate arrays, run in opposite directions and averaged, with decay applied as a separate step. It supplies $\Delta t_d = 30$ s and $\Delta t_m = 5$ s.

This option is not selected. It is specified because it accounts for the auxiliary u and v arrays, the one-compartment halo and the invariant relating them to the concentration.

The u sweep runs forward, increasing in both indices, and takes the neighbours above and to the left at the new level, those below and to the right at the old:

$$u_{i,j}^{n+1} = a\, u_{i,j}^{n} + b\left(u_{i-1,j}^{n+1} + u_{i+1,j}^{n}\right) + c\left(u_{i,j-1}^{n+1} + u_{i,j+1}^{n}\right).$$

The v sweep runs in reverse, decreasing in both indices, and takes the neighbours below and to the right at the new level, those above and to the left at the old. The coefficients are common to both:

$$\lambda = D\,\Delta t_d\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right), \qquad a = \frac{1-\lambda}{1+\lambda}, \qquad b = \frac{D\,\Delta t_d}{\Delta y^2\,(1+\lambda)}, \qquad c = \frac{D\,\Delta t_d}{\Delta x^2\,(1+\lambda)}.$$

Reading values already written in the same pass gives better stability than a plain explicit method; the opposite second sweep and the averaging cancel the directional bias. The scheme is that of Barakat and Clark; b and c follow the code's index convention. The cutoff (M1.3, Normalisation and the cutoff) is applied within each sweep at each compartment.

The concentration is recovered with decay applied as a separate step:

$$u \mathrel{*}= e^{-k_{\deg}\Delta t_d}, \qquad v \mathrel{*}= e^{-k_{\deg}\Delta t_d}, \qquad C = \frac{u+v}{2}.$$

An averaging variant without the decay factor is used for shedTNFR2.

The invariant C = (u + v)/2 must hold whenever a sweep begins, since the sweeps read u and v, not C. A write to C alone would be invisible to the next sweep and then overwritten by the average. The grid's field accessors therefore increment all three arrays together, and a direct write must not bypass them. Under the spectral option the arrays are maintained and never read.

Field handling

The derived assignment (M1.3, Derived assignment) is fused into the CCL2 averaging step.

The TNF sweep is the halo-reading variant, selected at compile time. The halo is never written and holds zero, so it realises a zero-concentration boundary, consistent with the spectral option and not periodic. CCL2 and IL-10 take the same halo-reading sweep, and so the same boundary.

The anti-TNF species and the antibiotics use no-flux sweeps. These accumulate the coefficient of each neighbour outside the domain and divide by $1 - \mathrm{coef}$, which renormalises the stencil rather than reflecting the centre value.

The explicit solver neither advances nor copies oxygen (its oxygen block is compiled out). With N_d = 20, an even count, the starting buffer survives with twenty applications of each oxygen routine per cellular timestep; accumulated consumption and supply equal those of the spectral option, though the substep size differs (S7.10).

Constraints

The accuracy heuristic uses $\kappa = 2.0$ (M1.2). The selection routine also imposes a hard ceiling of 60 s on this option's Δt_d, failing at startup with a diagnostic; the spectral option has no such ceiling.

M1.5 Alternative option: Crank-Nicolson ADI

An implicit alternating-direction scheme. It supplies $\Delta t_d = 60$ s and $\Delta t_m = 3$ s.

This option is not selected.

It is selected by value 7 and exists only in the two-dimensional build (M1.1). Its pair gives N_d = 10 and N_m = 20 and satisfies every synchronization condition (M1.2). It shares the spectral accuracy factor $\kappa = 10^4$. Its solver alone takes constructor arguments beyond the grid dimensions: the compartment length and Δt_d.

Its implicit matrices are built from the diffusivity once, at construction; the decay constant is read at each solve; and the solver requires a square grid (M7.6).

Its constructor also builds an oxygen system from `O2_diffusivityO2`, so that parameter is read under value 7; the system is never applied (Appendix B).

The smoother runs only under the spectral options, so this option has none.

M1.6 Parameters

Identifier	Value	Range	Units	Meaning
`Diffusion_FFT_timestepDiffusion`	60		s	Δt_d, spectral options
`Diffusion_FFT_timestepMolecular`	6		s	Δt_m, spectral options
`Diffusion_FFT_numSmootherSteps`	0		integer count	Smoother iterations per call (M1.3, The smoother)
`Diffusion_FFT_timestepDiffusionSmoother`	1.2		s	Smoother timestep Δt_s; loaded at startup; never consulted at zero iterations
`Diffusion_ADE_timestepDiffusion`	30		s	Δt_d, explicit option; loaded at startup and used by the selection routine at construction (ceiling check); not used for the run unless selected[1]
`Diffusion_ADE_timestepMolecular`	5		s	Δt_m, explicit option; loaded at startup; not checked; not used unless selected
`Diffusion_CNADI_timestepDiffusion`	60		s	Δt_d, Crank-Nicolson option; loaded at startup; used only if selected
`Diffusion_CNADI_timestepMolecular`	3		s	Δt_m, Crank-Nicolson option; loaded at startup; used only if selected
`Core_diffusivityTNF`	5.2×10^{-8}		cm^2/s	TNF; sole input to the accuracy heuristic (M1.2)
`Core_diffusivityChemokines`	5.2×10^{-8}		cm^2/s	CCL2, and through it CCL5 and CXCL9
`Core_diffusivityIL10`	sampled	$[4 \times 10^{-8}, 6 \times 10^{-8}]$	cm^2/s	IL-10; at most about 1.15 times the TNF diffusivity; fixed for a run

[1] A value above 60 s fails construction on every run and every load, whichever option is selected (M1.1). The selected pair must satisfy the conditions of M1.2.

`Core_diffusivityIL10` is the only sampled quantity in this module. None of the parameters in the table declares a default, so omitting one from the parameter file fails startup; all are read at startup (spine 7.0, Resolution).

Fixed structural constants:

- `DIFFUSION_CUTOFF` = 10^{-12} molecules per compartment (M1.3, Normalisation and the cutoff).
- Normalisation factor 1/(4N), 1/360000 on the default grid (M1.3, Normalisation and the cutoff). Unlike the other constants, it depends on the transform convention.
- Accuracy factor κ: 2.0 (explicit); 10^4 (spectral and Crank-Nicolson) (M1.2).

- `MOLECULAR_ACCURACY` = 30 s, the ceiling on Δt_m (M1.2).
- Explicit-option ceiling on Δt_d: 60 s (M1.4).
- `AGENT_TIME_STEP` = 600 s, the numerator of N_d; `SECONDS_PER_TIMESTEP` = 600 s, derived from `SECONDS_PER_DAY` = 86400 and `TIME_STEPS_PER_DAY` = 144 and used in the divisibility checks (M1.2).
- `MICROCMPT_LENGTH_CENTIMETERS` = 20×10^{-4} cm (Δx); `MICROCMPT_FACE_AREA_CENTIMETERS_SQUARED` = 4×10^{-6} cm^2 (Δx^2); $\ell = 2/\Delta x^2 = 5 \times 10^5$ cm^{-2}. Shared physical constants, including `NAV` and `VOL` (used by the drug permeability limit and the M7.7 binding blocks), are owned by spine 2.1.
- shedTNFR2 decay constant: 0, a literal (M1.3, The propagator).

None of these can be supplied externally, so a reimplementation must reproduce each, except the normalisation factor, which follows the transform convention.

Owned elsewhere and read here: `Core_kDeg`, `Core_ChemokinekDeg` and `Core_Ikdeg` (S6 parameter table); `Core_diffusivityTGFBActive`, `Core_degRateTGFBActive` and `Core_degRateTGFBInactive` (M4.9); `Core_Mac_dCCL2`, `Core_Mac_dCCL5` and `Core_Mac_dCXCL9`, read only as the two quotients (M1.3, Derived assignment; S6).

Read but without effect by default: `Molecular_TNF_diffusivityShedTNFR2` and `Molecular_TNF_kOn2` (M3 disabled); the `Mod_*_diffusivity` and `Mod_*_degRateConst` entries of the six anti-TNF species, both dosages, the Etanercept and Infliximab `kOn` and `kOff`, and `Mod_dosageStartTime` (M7 disabled); `O2_diffusivityO2` and `O2_degRateConstO2`, read and stored, but read only by the compiled-out oxygen path and, under value 7, by the unused Crank-Nicolson oxygen system. `Molecular_TNF_KD2` is inert in the binding sweep (M1.3, Post-solve binding sweep) but active in the coarse TNFR2 sink (S6). The six `Mod_*_minDiffusivity` entries are declared and not read by the solver (M7.6).

M1.7 Interfaces

With	Provides or receives	See
Spine Element 3	Provides Δt_d, Δt_m, N_d and N_m to the schedule, set at startup; the field advance sits between the two split half-steps	M1.1, M1.2
Spine 2.1	Links to M1 for the synchronization conditions; owns the shared physical constants and the significant-figure truncation	M1.2, M1.3 (planning strategy)
Spine 2.2	Owns the halo, container sharing, the exchange parity and the agent–field boundary difference; this module supplies the field side	M1.3 (boundary conditions, non-transformed fields and grid buffering)
Spine 2.4	Owns the field inventory	M1.3 (fields presented)
Spine 3.3	Owns the Strang splitting, which depends on the validated parity of N_m and on one field advance between the half-steps	M1.2

With	Provides or receives	See
Spine 7.0	Owns the omitted-parameter rule (Resolution), the range-check rule (Validity ranges) and the computed-parameter derivations (Computed parameters)	M1.6
S1	Receives the derived CCL5 and CXCL9 for chemotaxis, and the zero-concentration field boundary	M1.3 (boundary conditions, derived assignment); S1.3
S3	Receives the substep counts	M1.1, M1.2; S3.13
S5	Caseum secretion writes CCL2, CCL5 and CXCL9 at a quarter of the macrophage rates in the half-step	M1.3 (derived assignment); S5.6
S6	Receives the advanced TNF, CCL2 and IL-10 fields, which S2, S3 and S8 read through it; owns the derived-field relation, the cytokine decay constants and the secretion rates	M1.3 (propagator, derived assignment); S6.14
S7	Oxygen is neither advanced nor copied; only the history of the grid object current at the end of the loop survives (S7.1, S7.6)	M1.3 (fields presented, non-transformed fields and grid buffering); S7.1, S7.6, S7.10
S8	Receives the derived CCL5 and CXCL9 for recruitment	M1.3 (derived assignment)
M2	Option B's shared predicates read the derived chemokine fields (M2.14, S8.5)	M1.3 (derived assignment); M2.14
M8, M9	No coupling	—
M3	Its TNF receptor flag gates the shedTNFR2 transform, smoothing and dissociation; M3.3 specifies the shed TNFR2 field	M1.3 (fields presented, post-solve binding sweep); M3.3
M4	Both TGF-β fields are processed whether or not M4 is enabled; collagen is not copied forward. M4.9 tables the TGF-β diffusivity and decay constants	M1.3 (fields presented, propagator, non-transformed fields and grid buffering); M4.4, M4.9
M5	Supplies the grid dimensions, which fix the transform size and every propagator; Crank-Nicolson is two-dimensional only; a cloned three-dimensional solver uses no-flux plans with a sine transform on z (M5.10)	M1.1, M1.3 (propagator, normalisation and the cutoff); M5.14
M6	Each drug builds its own propagator from the Δt_d supplied by this module, using this solver's propagator routines, and carries its own grid pair and exchange; drug checks at M6.4	M1.2, M1.3 (propagator, non-transformed fields and grid buffering); M6.4, M6.5
M7	The immunomodulation gate controls the anti-TNF transforms and smoothing; the dosages gate the binding blocks; the binding chemistry, the Etanercept sign and the drug-field solver details are in M7	M1.3 (fields presented, post-solve binding sweep); M7.2, M7.6, M7.7

With	Provides or receives	See
Appendix B	Indexes `--diffusion` (default 5) and the inert parameters	M1.1, M1.6