行政院原子能委員會

放射性物理管理局

核二廠乾式貯存設施設計壽命期間熱傳行 為分析

期末報告

財團法人核能與新能源教育研究協進會

施純寬

104FCMA011

中華民國 104 年 12 月 07 日

摘要

本研究係針對核二廠乾式貯存設施於設計壽命期間的熱流行為進行研究, 以期透過計算流體力學分析技術,瞭解核二乾貯系統各元件與密封鋼筒於不同 貯存條件下之溫度分布與可能之應力腐蝕趨勢,並藉以建立一預估工具以供管 制單位使用。經過年計畫的執行,本團隊已依計畫原訂之工作時程,完成模式 發展、參數靈敏度分析與設計壽命期間之熱流分析、應力腐蝕分析等工作項目 以及空氣流道對流場之影響探討。參數靈敏度測試結果顯示,相較於環境溫度 與燃料束類型,裝載配置與內部之總熱負載將是影響整體熱流特性之主要關鍵 參數。因此本研究選擇以平均配置與集中擺置兩種模式進行核二廠設計壽命期 間之熱流與應力腐蝕現象分析。研究結果更顯示,空氣流道內的溼度與活化能 係一組趨勢相反之溫度函數,故TSC表面預期將不會輕易因應力腐蝕而裂穿。 而基於上述結果所建立的熱流預估工具發展與 TSC 表面流場對鹽霧附著進行 技術探討更顯示,自然對流所具有的液滴挾帶能力有限,且其液滴的分佈往往 會被空氣流道的複雜外型阻隔而減低,故可推測在實際乾貯系統之表面鹽類沉 積速度,將遠小於一般露天測試之結果。

關鍵字:核二廠乾貯系統、設計壽命期、熱流分析、應力腐蝕破裂

ABSTRACT

The goal of study is to investigate the thermal hydraulic behavior of Kuosheng dry storage system during the design lifetime. A computational fluid dynamics (CFD) technology has been employed to develop a useful analysis method for this study, which can not only be used to evaluate the thermal behavior but also to examine the stress corrosion crack (SCC) induce by the deliquescence Chloride on the TSC. Through hard work in this year, the model developing, parameters sensitivity test, thermal and SCC analysis and internal flow investigation works have been completed by the research team on schedule. The result shows that the influence induced by the fuel pattern is the major parameter o for the thermal behavior of KSDSS. For this reason, a central peak and uniformed pattern are selected for the investigating during the design lifetime of KSDSS. The result also indicates that the active energy and possibility of the SCC have reversed relationship to the temperature distribution on the TSC shell. For this reason, the TSC shell will not easily occurred in the design lifetime because the entrainment mechanism will be limited by the droplet size and the complex geometries of the air-channel, which means that the realistic Chloride deposition rate will reasonable lower than the result from open-air test.

Keywords : KSDSS, TSC, SCC, design lifetime

目錄

摘要	.i
ABSTRACT	ii
目錄i	ii
圖目錄	V
表目錄vi	ii
一、研究背景與目的	1
二、統御方程式與數值模式	4
2.1 統御方程式	4
2.2 數值模式	5
2.2.1 DO 熱輻射模式	5
2.2.2 紊流模式	7
2.2.3 自然對流模式	8
三、核二廠乾貯系統幾何資料蒐集	9
3.1 系統幾何	9
3.2 材料性質1	7
3.2.1 流體熱傳性質[3]1	7
3.2.2 固體熱傳性質1	8
3.3 燃料衰變熱計算模式建立2	21
3.4 燃料軸向功率分佈2	25
3.5 燃料配置2	28
四、分析模式發展、邊界條件定義與案例選擇	0
4.1 模式發展與網格生成	0
4.2 邊界條件設定	6

4.3 靈敏度測試與案例選定37
4.3.1 環境溫度之影響評估
4.3.2 燃料類型之影響評估40
4.3.3 燃料運轉歷史與佈局之影響評估45
五、設計壽命期間之熱流特性及空氣流道流場研析
5.1 核二廠乾式貯存設施密封鋼筒設計壽命期間元件溫度分布模擬研析
5.1.1 設計壽命期間元件溫度分布研析成果匯整57
5.2 乾貯系統設計壽命期間各元件溫度趨勢預估工具之建立59
5.2.1 溫度趨勢預估工具發展成果匯整60
5.3 核二乾式貯存設施密封鋼筒設計壽命期間之熱流特性與應力腐蝕關
係研析65
5.3.1 熱流特性與應力腐蝕關係研析成果匯整
5.4 乾貯系統空氣流道之流場研析與鹽類附著之可能位置分析71
5.3.1 流場研析與鹽類附著可能位置之研析成果匯整73
六、結論
參考文獻
附錄一
附錄二
附件三

置	1 燃料提籃聯結銷	.10
圖	2 燃料提籃結構	.11
圖	3 密封鋼筒	.12
圖	4 混凝土內襯	.13
圖	5 混凝土護箱	.14
圖	6 混凝鋼質底座與通氣口弧長	.15
圖	7 不同衰變熱估算方法之差異	.23
圖	8 核二各式可能貯存燃料之軸向功率分佈曲線	.26
圖	9、GE-8X8-2、GE9B 燃料之軸向功率分佈	.27
圖	10、ATRIUM-10 燃料軸向功率分佈	.27
圖	11、核二乾貯系統燃料佈局配置規劃試算表	.29
圖	12、核二乾貯系統幾何示意圖	.32
圖	13、核二乾貯系統幾何示意圖(a)整體護箱之網格分佈(b)TSC 與內]部
	提籃之網格分佈	.33
圖	14、核二乾貯提籃、中子屏蔽板與燃料束示意圖	.34
圖	15、核二乾貯系統整體網格分佈	.35
圖	16、核二乾貯系統各元件溫度受環境影響之指標	.39
圖	17、核二乾貯系統裝載 GE8X8 時之系統溫度分佈	.41
圖	18、核二乾貯系統裝載 ATRIUM-10 時之系統溫度分佈	.42
圖	19、核二乾貯系統內部對流現象	.43
圖	20、核二乾貯系統貯存不同燃料元件時TSC 之溫度分佈差異	.44
圖	21、不同燃料佈局之案例(a)TYPE-I、(a)TYPE-II、(a)TYPE-III。	47
圖	22、不同燃料佈局案例之整體溫度(°C)分佈(a)TYPE-I、(a)TYPE-I	I١

圖 23、不同燃料佈局案例於燃料有效發熱段頂端之橫截面溫度(°C)分 圖 28、不同初始裝載之燃料於分析貯存期間之衰變熱變化趨勢.......55 圖 29、貯存2年後之整體溫度分佈(a) 平均配置 (b)TYPE-III 配置..55 圖 30、貯存 100 年後之整體溫度分佈(a) 平均配置 (b)TYPE-III 配置 圖 31、平均配置下之 TSC 溫度在不同貯存年份下之變化趨勢........56 圖 32、TYPE-I 配置下之 TSC 溫度在不同貯存年份下之變化趨勢.....57 圖 33、不同元件經無因次化後的指數適配結果......61 圖 34、初始裝載功率與貯存年限之關係式......61 圖 35、燃料溫度與功率之關係式......62 圖 36、提籃溫度與功率之關係式......62 圖 38、內襯溫度與功率之關係式......63 圖 39、VCC 溫度與功率之關係式64 圖 41 台北近年之相對溼度......67 圖 42 304 不銹鋼於不同溼度與溫度條件下發生 SCC 之分佈範圍....67

啚	46、核二乾貯系統空氣流道內溫度及流場分佈	.74
圖	47、核二乾貯系統底座焊件工程圖	.75
圖	48、核二乾貯系統底座附近流場分佈圖	.76
圖	49、核二乾貯系統底座與空氣流道速度分佈	.76
圖	50、整體空氣流道速度分佈	.77
圖	51、不同粒徑在流道內的分佈差異	.78
圖	52、核二乾貯系統空氣流道底部流體蓄積現象	.78

表	1 核二乾貯系統重要尺寸列表1	16
表	2 氦氣熱傳性質1	17
表	3 空氣熱傳性質	17
表	4 不銹鋼(304 及 304L)熱傳特性1	18
表	5 碳鋼(A-36、A-533 及 SA-588)熱傳特性	18
表	6 鉛屏蔽熱傳特性	19
表	7 鋁合金(6061-T651)熱傳特性1	19
表	8 鋯合金熱傳特性1	19
表	9 燃料丸(UO2)熱傳特性2	20
表	10 混凝土熱傳特性	20
表	11 各種不同衰變熱估算模式之計算結果2	24
表	12 影響設計壽命期間熱流特性之參數	37
表	13 核二乾貯系統裝載不同燃料時之元件最高溫度(℃)	43
表	14 核二乾貯系統裝載不同配置時之元件最高溫度(℃)	50
表	15 核二乾貯系統均勻擺置時,經不同貯存年份下之元件最高溫度	
	(°C)	58
表	16 核二乾貯系統於密集擺置時,經不同貯存年份下之元件最高溫	度
	(°C)	58

一、研究背景與目的

我國核一與核二廠係美國奇異公司之沸水式反應器(Boling Water Reactors, BRWs),其爐心燃料束依系統設計熱功率之不同而分別具有為408 與624 束燃 料。依據此二電廠之爐心佈局規劃,每次大修時需退出1/3 左右之燃料,剩餘 燃料在加入新的燃料束並經過挪移以滿足下一循環之設計需求。也因此核一二 廠四座機組運轉至今,已退出總數逾萬之用過燃料束,即使兩座電廠的燃料池 已於早年依據原能會所頒佈之「核能電廠用過燃料池貯存格架改裝安全分析報 告審查規範」[1]進行格架改裝工程而增加了貯存空間,但仍無法滿足現有核 電廠運轉40年之貯存需求。因此台電公司參照歐美日等先進國家之成熟做法, 於核電廠現有廠區內規劃並興建乾式貯存設施,再將目前貯存於燃料池中多年、 已充份冷卻之用過燃料移往乾式貯存設施進行乾式貯存,以確保電廠燃料池具 有充足空間來滿足原能會之運轉要求。目前核一廠乾式貯存設施[2]已完成功 能驗證測試,而核二廠乾式貯存設施[3]則已完成安全審查。

值得注意的是,上述兩套乾貯系統於最初進行安全分析時,係以乾式貯存 護箱之最大設計熱負載(核一為14kW,核二為17kW)之保守條件進行熱流評估, 以求得較為保守之系統元件溫度以及容許處置時間。但實際裝載作業時,台電 係規劃較低熱負載之配置來進行燃料束的裝填與貯存。此舉使得系統初始之熱 流餘裕已明顯高於原安全分析規劃;此時而若再將用過核子燃料其衰變熱將隨 時間的增長而減弱納入考慮,將使得系統各元件的溫度表現與最初分析結果存 在明顯之落差。考量長期貯存期間之護箱劣化,以及未來用過核子燃料仍須要 傳送到處置場進行最終處置之雙重前提下,管制單位確有必要有對設計壽命期 間的系統熱流特性進一步地掌握與確認。

因此,行政院原子能委員會放性物料管理局(簡稱物管局)於103年已針對 核一廠乾式貯存設施於設計壽命期間熱傳行為[4]進行研究,並瞭解到設計壽 命期間內部熱負載與元件溫度、密封鋼筒溫度分佈之相關性。其研究結果指出, 核一廠乾式貯存設施於設計壽命期間,系統各元件之溫度與其分佈將因為條件 的不同而有明顯的差異,並進而使系統具有不同程度之劣化趨勢;例如高熱負 載條件下,乾貯系統元件的溫度將會較高,並促使TSC 表面之相對溼度較低, 故系統環境將不致處於可能發生劣化之環境條件下,反之略低的熱負載則將使 劣化效應逐漸顯現,直至活化能再次因為熱負載的大幅衰退而降低而趨近於零 為止。

有鑑於各式乾貯系統之熱流特性,將因為不同設施之設計方式與裝載配置 差異而呈現不同之狀況,故其上述計畫所獲之核一廠熱流特性,將無法直接應 用於我國核二廠乾貯系統之中。因此,本研究擬藉由計算流體力學 (Computational Fluid Dynamics, CFD)分析技術[5],針對核二廠乾貯系統於設計 壽命其間之各元件面溫度分佈進行探討,以協助管制單位釐清該系統於設計壽 命期間之熱流特性。另一方面,由於空氣中的鹽類往往係因海水或是雨水沖刷 土壤後,以微小液滴方式由風挾帶並進入乾貯系統之空氣流道,並隨著氣體於 空氣流道內的轉折所造成流場衝擊而附著於特定區域之表面。因此本研究擬著 利用 CFD 分析技術探討空氣流道內之流場特性,並瞭解氣體中所挾帶之鹽類 可能之附著位置,以釐清鹽類附著於空氣流道兩側(內襯及 TSC 表面)之特性。

基於上述原因,本研究擬於計畫執行期間,針對我國核二廠乾貯系統之設 計,進行設計壽命期間之熱流特性研析外,除了希望與既有核一廠設計壽命期 間之分析結果進行全面性的比較與探討,更希望透過對乾貯系統之空氣流道流 場分佈與鹽類可能附著狀況進行研析,以精進乾式貯存設施十年再評估時可能 需具備的老化管理與管制所需之審查能力。

本研究預期可瞭解核二廠乾貯設施於設計壽命期間之熱流特性,並進一步 匯整所得之經驗式與無因次結果,並將其發展為一套完整之元件溫度預估工具, 以協助相關單位作為未來管制與學術研究之基礎。

依原計畫書之規劃,本計畫之主要工作項目可概分如下:

1.核二廠乾式貯存設施密封鋼筒設計壽命期間元件溫度分布模擬研析

2.核二廠乾式貯存設施密封鋼筒設計壽命期間熱流特性與應力腐蝕關係
 研析

3.乾貯系統設計壽命期間各元件溫度趨勢預估工具之建立

4.乾貯系統空氣流道之流場研析與鹽類附著之可能位置分析

5.撰寫研究計畫成果報告

而本期末報告係依計畫預定時程所述,針對全年所執行之核二乾式貯存設

施於設計壽命期間之元件溫度分布、應力腐蝕關係等相關與空氣流道內流場研析等成果進行描述。

二、統御方程式與數值模式

2.1 統御方程式

本研究以係採用與美國 NRC NUREG-2174[6]相同,並已多次應用於我國 乾貯相關熱流分析作業[7-11]之商用計算流體力學軟體 FLUENT 為基礎。在進 行一系列之模式建立、網格生成與測試、數值模式測試與比對、假設、簡化以 及邊界條件靈敏度測試等工作後,本研究順利完成乙套符合本分析需求之分析 技術建立,並藉以完成核二乾貯系統在設計壽命期間之熱流特性、應力腐蝕可 能性分析。基於設計壽命期間之熱流特性主要係以針對乾貯系統之混凝土護箱 於正常狀態下長期貯存,故其熱流特性可由以下之穩態統御方程式[5]來加以 描述:

質量守恆方程式

$$\nabla \cdot \left(\rho \vec{v} \right) = 0 \tag{1}$$

其中,

 ρ 為流體密度;

而ī為速度向量。

動量守恆方程式

$$\nabla \cdot \left(\rho \vec{v} \vec{v}\right) = -\nabla p + \nabla \cdot \left(\vec{\tau}\right) + \rho \vec{g} + \vec{F}$$
⁽²⁾

其中, p為靜壓力, $\rho \vec{g}$ 為體積力項;

$$\vec{F} \hspace{0.1cm} \begin{array}{l} \hspace{0.1cm} \bar{F} \hspace{0.1cm} \hspace{0.1cm} \hspace{0.1cm} \bar{a} \hspace{0.1cm} g \hspace{0.1cm} \hspace{0.1cm} h \hspace{0.1cm} \bar{\tau} \hspace{0.1cm} \hspace{0.1cm} \hspace{0.1cm} \bar{b} \hspace{0.1cm} \bar{c} \hspace{0.1cm} h \hspace{0.1cm} \bar{c} \hspace{0.1cm} \bar{c}$$

μ為流體黏滯係數;

能量守恆方程式

 $\nabla \cdot [\vec{v}(\rho E + p)] = \nabla \cdot (\kappa_{eff} \nabla T) + S_h \qquad (4)$ 其中, E 為單位質量之總能量; h 為焓值; p 為靜壓力; v 為速度向量; κ_{eff} 則為等效熱導率: $k_{eff} = k + k_i \qquad (5)$ 其中,

k 為物質本身之分子熱導率;

k, 則為流場紊流所致之熱導率。

2.2 數值模式

2.2.1 DO 熱輻射模式

上述統御方程式雖可描述熱流現象中的流場以及傳導對流等熱傳機制,但仍 無法描述熱輻射機制對此系統之影響。在此平行驗證工作中採用於前一期研究中 已充份校驗之熱輻射模擬模式-DO模式。DO模式除已於許多乾貯相關的研究中 被加以應用外,本研究團隊更有充份的使用經驗與嚴僅的校驗程序來確保分析模 式之精確性。

DO 模式主要乃是求解輻射傳輸方程式(Radiation Transport Equation)來計算

有限體積法中各網格之熱輻射效應,其式可以寫為:

$$\nabla \cdot (I_{\lambda}(\vec{r},\vec{s})\vec{s}) + (a_{\lambda} + \sigma_{s})I_{\lambda}(\vec{r},\vec{s}) = a_{\lambda}n^{2}I_{b\lambda} + \frac{\sigma_{s}}{4\pi} \int_{0}^{4\pi} I_{\lambda}(\vec{r},\vec{s}') \Phi(\vec{s},\vec{s}') d\Omega' ; (6)$$

其中, $I(\vec{r},\vec{s}) \triangleq \vec{r}$ 點沿著 \vec{s} 方向之總輻射強度其式可以寫為:
 $I(\vec{r},\vec{s}) = \sum_{k} I_{\lambda_{k}}(\vec{r},\vec{s}) \Delta\lambda_{k} ; (7)$
 λ 為波長;
 a_{λ} 為吸收系數;
 $I_{b\lambda}$ 為黑體輻射強度;
 σ_{s} 為散射系數;

n 為折射率(refractive index)。

2.2.2 紊流模式

由於乾貯系統所包含的傳送護箱與貯存用混凝土護箱,其高度皆在5m以上, 因此高度所造成的自然對流動強度將促使內部對流呈現紊流流場。為求能精確的 掌握紊流所造成的熱水流效應,本研究擬在相關分析採用適用性已受認同的 kω作為紊流模擬之數值模式, k-ω模式之方程式可以表示為:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_{i}}(\rho \kappa u_{i}) = \frac{\partial}{\partial x_{j}}\left(\Gamma_{k}\frac{\partial k}{\partial x_{j}}\right) + \tilde{G}_{k} - Y_{k} + S_{k}$$

$$\frac{\partial}{\partial t}(\rho \omega) + \frac{\partial}{\partial x_{i}}(\rho \omega u_{i}) = \frac{\partial}{\partial x_{j}}\left(\Gamma_{\omega}\frac{\partial \omega}{\partial x_{j}}\right) + \tilde{G}_{\omega} - Y_{\omega} + S_{\omega}$$
(8)
$$(9)$$

其中, k 為紊流動能, 而 O 則為消散率; \tilde{G}_{k} 為紊流動能梯度, \tilde{G}_{ω} 為消散 率梯度; Y_{k} 與 Y_{ω} 分別為k與 O 因為紊流所造成的消散量; S_{k} 與 S_{ω} 分別為k方 程式與 O 方程式之源項。 Γ_{k} 與 Γ_{ω} 則為等效擴散項,上述變數之定義可以表示 為:

$$\Gamma_k = \mu + \frac{\mu_t}{\sigma_k} \quad ; \tag{10}$$

$$\Gamma_{\omega} = \mu + \frac{\mu_t}{\sigma_{\omega}} \quad ; \tag{11}$$

其中, μ 為流體黏滯係數, $\sigma_k 與 \sigma_\omega$ 為紊流普蘭特常數(turbulent Prandtl number)而, μ_i 則為:

$$\mu_t = \alpha^* + \frac{\rho k}{\omega} \quad ; \tag{12}$$

其中, 紊流阻尼 α^* 可以定義為

$\alpha^* = \alpha_{\infty}^* + \left(\frac{\alpha_0^* + \operatorname{Re}_t / \operatorname{R}_k}{1 + \operatorname{Re}_t / \operatorname{R}_k}\right) ;$		(13)
而, α^*_{∞} 為阻尼;		
$\operatorname{Re}_{t} = \frac{\rho k}{\mu \omega}$;	(14)	
$\mathbf{R}_{k}=6;$	(15)	
$\alpha_0^* = \frac{\beta_i}{3} ;$	(16)	
$\beta_i = 0.072$.	(17)	

2.2.3 自然對流模式

在 FLUENT 中的自然對流主要乃是藉由流體密度差以及重力場的設定而達成動量方程式中體積力項(ρg)的計算。由於重力場乃是一常數,因此如何模擬流體密度之變化即是建立自然對流模擬之重要參數。

在 FLUENT 中,用以設定流體密度變化的機制可以分為三種,其一為直接 設定法,直接於材料性質表中,將材料性質設定為溫度之函數;或利用布希尼克 (Boussinesq)近似法與理想氣體方程式來進行密度之計算。有鑑於採用布希尼克 近似法來進行求解有助於增進求解之效率與收斂性,故本研究遂選用此法進行流 體性質與體積力之設定。

8

三、核二廠乾貯系統幾何資料蒐集

本章主要用以記載與核二系統幾何與材料性質之資訊,以供後續執行模型建 立與邊界條件建構作業之用。其中,系統幾何資料主要以進行分析模式建立,而 材料性質則是作為建立合理分析邊界條件之依據。

3.1 系統幾何

依據台電已審查通過之核二乾貯系統安全分析報告[3]可以得知,核二乾貯 系統為一可容納87束BWR用過核子燃料之系統,其單一燃料套筒為一外徑6.59 吋之方型套管,內嵌中子吸收板與護保片後其內徑為5.86 吋。此外值得注意的 是,核二系統採用交錯式組合法進行提籃結構的設計,藉由四個相臨的套筒藉由 螺栓扣鎖(圖1)而額外組成一個容納空間(如圖2與所示);此一配置重覆的出現 於整個系統中,直到提籃最外側再以護板加固,進而組合出可容納89 個燃料組 件的空間,其中由於2 個燃料方管位置係用以裝設抽、注水設備而不用於存放燃 料,是以整個系統之最大貯存容量為87束。

這個組合後的提籃將會置於一外徑為72吋,高度為191.8吋之密封鋼筒內(見 圖 3);鋼筒的上蓋為9吋厚而下底則為2.75吋厚,鋼筒厚度則為0.5吋厚。在 將用過燃料置入筒內之燃料提籃後,作業人員將會對上蓋施以銲接以達密封之目 的。另外,考量本研究無需針對傳送作業進行分析,故於此報告中不再就此傳送 護箱進行贅述。

完成裝載之密封鋼筒,將會置於一具有外徑約為136 吋,高約225.3 吋之混 凝土護箱(圖 5)進行長期貯存;其內部有一鋼質底座(圖 6)以乘載密封鋼筒之重 量。值得注意的是,此一底部有四個通氣口供空氣流入護箱內,並令空氣得以沿 密封鋼筒表面與護箱內襯(圖 4)之間約3.75 吋之間隙向上流動而達冷卻之目的。 最後,經密封鋼筒表面加熱的空氣將會經由混凝土護箱上方的出氣孔排出。以上



圖 1 燃料提籃聯結銷



圖 2 燃料提籃結構



圖 3 密封鋼筒



圖 4 混凝土內襯



圖 5 混凝土護箱



圖 6 混凝鋼質底座與通氣口弧長

系統組件	元件	特徵尺寸	數值
		直徑	72
密封鋼筒		長度	191.8
		厚度	0.5
		長度	179.5
做业担贷		直徑	70.76
然 叶灰監		總貯存量	87
		套筒數	45
	內襯	厚度	1.75
		外徑	83
混凝土護箱	通氣孔	高度	2
		寬度	136
	混凝土	厚度	26.5
傳送護箱	內殼板	厚度	0.75
	外殼板	厚度	1.25
	鉛屏蔽	厚度	3.2
	中子屏蔽	厚度	2.25
	屏蔽門	厚度	5

表 1 核二乾貯系統重要尺寸列表

3.2 材料性質

核二廠乾貯系統各元件之熱傳特性經蒐集後可以分為流體與固體兩大類,以 下將分析詳述所蒐集到之材料性質與其處理後之結果。

3.2.1 流體熱傳性質[3]

考量本計畫著眼於該系統於設計壽命期間之長期貯存狀況,故其分析作業可 由正常案例作為評估之基礎,故分析時只需考慮長期貯存時內部所填充之氦氣以 及傳送與貯存過程中用以冷卻之空氣,而無需對傳送作業期間之冷卻水性質進行 處理。以上各式流體之材料性質如下表所示:

	溫度(K)							
	300	400	500	600	700	800		
DENSITY (Kg/m ³)	0.1625	0.1219	0.09754		0.06969			
Cp (J/Kg-K)	5193							
Viscosity (N-s/ m ²)	1.99e-5	2.43e-5	2.83e-5	3.20e-5	3.50e-5	3.82e-5		
Thermal Conductivity (W/m-K)	0.152	0.187	0.220	0.252	0.278	0.304		

表 2 氦氣熱傳性質

表 3 空氣熱傳性質

	溫度(K)	溫度(K)						
	300	400	500	600	700	800		
DENSITY (Kg/m ³)	1.1614	0.8711	0.6964	0.5804	0.4975	0.4354		
Cp (J/Kg-K)	1007	1014	1030	1051	1075	1099		
Viscosity ((N-s/m ²)	1.846e-5	2.301e-5	2.701e-5	3.058e-5	3.388e-5	3.698e-5		
Thermal Conductivity (W/m-K)	0.0263	0.0338	0.0407	0.0469	0.0524	0.0573		

3.2.2 固體熱傳性質

有鑑於本分析僅著眼於貯存狀況,故此節中僅就於貯存狀態下之各項材料, 如金屬類的碳鋼、不銹鋼、鉛、含硼鋁板、鋯合金等以及非金屬類的混凝土、鈾 燃料丸等進行記錄;所蒐集到之材料性質如下表所示:

劫值长州	温度(K)					
然时付任	296.72	352.27	463.38	546.72	657.83	
Conductivity (W/m-K)	15.05	16.09	18.00	19.21	20.76	
Density (Kg/m^3)	8016.37	7994.22	7949.93	7908.41	7858.59	
Specific Heat (J/Kg-K)	483.98	503.24	533.38	550.13	567.30	
Emissivity	0.36					

表 4 不銹鋼(304 及 304L)熱傳特性

表 5 碳鋼(A-36、A-533 及 SA-588)熱傳特性

劫使性财	溫 度 (K)						
然 序 行 住	296.72	352.27	463.38	518.94	630.05	685.60	
Conductivity (W/m-K)	41.36	41.59	41.88	41.01	38.77	37.54	
Density (Kg/m ³)	7861.36						
Specific Heat (J/Kg-K)	473.10						
Emissivity	0.8						

表 6 鉛屏蔽熱傳特性

劫值性趾	溫度				
然 停 行 住	371.45	477.55	578.15	605.35	
Conductivity (W/m-K)	33.87	31.69	25.12	20.93	
Density (Kg/m ³)	11380				
Specific Heat (J/Kg-K)	125.60				
Emissivity	0.28				

表 7 鋁合金(6061-T651)熱傳特性

劫庙はい	溫 度(K)					
	352.27	407.83	463.38	518.94	574.49	657.83
Conductivity (W/m-K)	171.30	174.00	176.28	176.28	176.28	176.28
Specific Heat (J/Kg-K)	962.94					
Emissivity	0.22					

表 8 鋯合金熱傳特性

劫康长州	溫 度 (K)				
然时付任	392	572	752	932	
Conductivity (W/m-K)	14.33	15.16	16.61	18.06	
Density (Kg/m ³)	6560.36				
Specific Heat (J/Kg-K)	301.44	309.82	318.19	330.75	
Emissivity	0.75				

表 9 燃料丸(UO2)熱傳特性

	溫 度 (K)				
烈	296.72	383.94	508.94	633.94	758.94
Conductivity (W/m-K)	7.89	7.20	5.75	4.90	4.40
Density(Kg/m ³)	1578	1716	1855	1965	2021
Specific Heat (J/Kg-K)	1658				
Emissivity	0.85				

表 10 混凝土熱傳特性

熱傳特性	溫 度(K)			
	296.72	352.27	407.83	
Conductivity (W/m-K)	1.89	1.85	1.79	
Density (Kg/m ³)	2243			
Specific Heat (J/Kg-K)	837.4			
Emissivity	0.9			
Absorptivity	0.6			

3.3 燃料衰變熱計算模式建立

有鑑於乾貯系統未來可能裝載之燃料,將因為燃料束自身之運轉歷史與退出 爐心之冷卻時間的差別,而具有不同的衰變熱,進而對系統元件之熱流特性與 TSC 表面溫度分佈造成影響。因此,本計畫針對核二廠燃料池內不同週期退出之 燃料,利用目前常見的幾個衰變熱模式;包括:ANSI/ANS-5.1-1979、ANS-5.1-1973 與 ASB 9-2 三種進行計算與比較,以確保所選用模式的合適性。

衰變熱計算結果顯示(表 11、圖 7), ASB9-2 模式於停機時間超過 200,000 秒後,其估算之衰變熱即會低於其它兩個預估模式,故預期未來直接提取核二燃料 束運轉歷史並估算設計壽命期間的熱流分析時, ASB 9-2 模式估算所得之衰變熱 功率將比其它兩種方式更為符合研究所需。

基於以上之比較,本研究即採用目前廣泛應用於核能界之 ASB 9-2 進行燃料 束衰變熱計算。ASB 9-2 於估算,衰變熱功率時,係將其分為分裂產物衰變熱與 重元素衰變熱功率兩部分:

分裂產物衰變熱:

$$\frac{P}{P_0}(\infty, t_s) = \frac{1}{200} \sum_{n=1}^{n=11} A_n \exp(-a_n t_s)$$
$$\frac{P}{P_0}(t_0, t_s) = (1+K) \frac{P}{P_0}(\infty, t_s) - \frac{P}{P_0}(\infty, t_0 + t_s)$$

其中, P/P_0 = 功率因子(fraction of operating power)

 t_0 = 運轉時間 (cumulative reacting operating time, seconds)

 t_s = 停機時間(time after shutdown, seconds) K = 不準度因子(uncertainty factor), 0.2 for $0 \le t_s < 10^3$ and 0.1 for $t_s \ge 10^3$

重元素衰變熱功率:

$$\frac{P(U-239)}{P_0} = 2.28 \times 10^{-3} C \frac{\sigma_{25}}{\sigma_{725}} [1 - \exp(-4.91 \times 10^{-4} t_0)] \times [\exp(-4.91 \times 10^{-4} t_s)]$$

$$\frac{P(N_p - 239)}{P_0} = 2.17 \times 10^{-4} C \frac{\sigma_{25}}{\sigma_{f^{25}}} \{ 1.007[1 - \exp(-3.41 \times 10^{-6} t_0)] \\ \times \exp(-3.41 \times 10^{-6} t_s) - 0.007[1 - \exp(-4.91 \times 10^{-4} t_0)] \\ \times \exp(-4.91 \times 10^{-4} t_s) \}$$

其中,P(U-239)/P₀ 為 U-239 的功率因子,而 P(Np-239)/P₀ 則 為 Np-239 的功率因子;

 t_0 = 運轉時間 (cumulative reacting operating time, seconds);

t_s =停機時間(time after shutdown, seconds);

C = conversion ration, atoms of Pu-239 produced per atom of U-235 consumed ;

 σ_{25} =effective neutron absorption cross section of U-235 ;

 $\sigma_{f25} \quad = \text{effective neutron fission cross section of U-235} ~ \circ$



圖7不同衰變熱估算方法之差異

Time	after	ANSI/ANS-5.1-1979	ANS-5.1-1973	ASB 9-2
shutdown (second)				
1.0E-1		0.0682	0.0675	0.0852
1.0E+0		0.0644	0.0625	0.07779
2.0E+0		0.0613	0.059	0.07357
4.0E+0		0.0569	0.0552	0.06898
6.0E+0		0.0540	0.0533	0.06622
8.0E+0		0.0519	0.0512	0.06415
1.0E+01		0.0501	0.050	0.06242
2.0E+01		0.0449	0.045	0.05635
4.0E+01		0.0398	0.0396	0.05023
6.0E+01		0.0369	0.0365	0.04703
8.0E+01		0.0349	0.0346	0.04477
1.0E+02		0.0334	0.0331	0.04291
2.0E+02		0.0291	0.0275	0.03638
4.0E+02		0.0255	0.0235	0.03025
6.0E+02		0.0234	0.0211	0.02763
8.0E+02		0.0219	0.0196	0.02610
1.0E+03		0.0208	0.0185	0.02302
2.0E+03		0.0176	0.0157	0.01925
4.0E+03		0.0141	0.0128	0.01521
6.0E+03		0.0125	0.0112	0.01331
8.0E+03		0.0115	0.0105	0.01228
1.0E+04		0.01074	0.00965	0.01160
2.0E+04		0.00891	0.00795	0.00962
4.0E+04		0.00751	0.00625	0.00759
6.0E+04		0.00668	0.00566	0.00663
8.0E+04		0.00612	0.00505	0.00610
1.0E+05		0.00571	0.00475	0.00574
2.0E+05		0.00458	0.00400	0.00465
4.0E+05		0.00359	0.00339	0.00349
6.0E+05		0.00306	0.00310	0.00293
8.0E+05		0.00272	0.00282	0.00261
1.0E+06		0.00249	0.00267	0.00239
2.0E+06		0.00182	0.00215	0.00179

表11各種不同衰變熱估算模式之計算結果

3.4 燃料轴向功率分佈

有鑑於不同燃料類型可能對系統元件之熱流特性與 TSC 表面溫度分佈造成 影響,進而改變系統於設計壽命期間之熱流分析結果。故本計畫在考量目前核二 尚未擬定裝載規劃之狀況,以核二安全分析報告中實際可能裝載之各式燃料之燃 料軸向功率模式進行邊界條件發展,除藉以探討各燃料束軸向功率分佈之影響外, 亦期望所分析之結果可以涵蓋整體待貯燃料束。

為了順利完成此分析技術之建立,本研究採用 FLUENT 程式前內建之使用 者自定函數(User Defined Function, UDF)[12]功能來進行技術發展,其主要機制係 利用 c 語言程式,針對各燃料束有效發熱段區間之網格座標,輸入一個隨高度而 變的軸向分佈係數(圖 9、圖 10)而達到所需之功率分佈模擬目的。

其中GE-8X8與GE-9B燃料之軸向功率因子(圖 9)f(y),可以被分段定義為:

$$\begin{cases} F(y) = 8y; 0 < y \le 0.15 \\ F(y) = 1.0; 0.15 < y \le 0.55 \\ F(y) = 0.95833; 0.55 < y \le 0.8 \\ F(y) = -5.6y + 0.95833; 0.8 < y \le 1.0 \end{cases}$$

而 ATRIUM-10 之軸向功率因子(圖 10) f(y),可以採多項式來加以表示: $F(y) = -61.266y^6 + 168.83y^5 - 187.298y^4 + 109.4y^3 - 38.146y^2 + 8.1456y$ + 0.3788

然而,由於 FLUENT 的熱源係以體積產生率(W/m³)為單位,因此若只將上 述的修正係數與平均分佈之衰變熱產生率帶入網格時,將導致整體熱源項將因為 前後端功率的減少而下降;而為了修正此數值處理上的缺失,最直接的方式即是 針對軸向功率分佈的整體積分值進行修正;例如,在一三角型的軸向功率分佈下, 其整體軸向功率係數僅為均勻分佈的 1/2;故在經由軸向位置求得其分佈係數後, 僅需再除上此整體軸向功率係數即可修正此一問題;而本研究中所選定之各式燃 料整體軸向修正係數則分別為:

 $\begin{cases} GE8 \times 8: 0.810417 \\ \text{ATRIUM} - 10: 1.0127 \end{cases}$



最後,經由上述作業程序而建立之軸向功率程式碼則如附件一所示。

圖 8 核二各式可能貯存燃料之軸向功率分佈曲線



圖 9、GE-8X8-2、GE9B 燃料之軸向功率分佈



圖 10、ATRIUM-10 燃料軸向功率分佈
3.5 燃料配置

除了燃料束之軸向功率有可能造成温度分佈的改變之外,燃料於提籃內的配 置狀況亦會對內部流場造成不同的影響。因此本計畫於執行後續的熱流分析前, 先行針對燃料配置分析作業,建立完整之分析技術,除藉以滿足本計畫中之各式 分析需求外;更希望藉此分析技術,為未來核二實際裝載配置影響之評估與後續 管制基礎提供必要之分析能力。

本工作項目除利用 FLUENT 所提供之 UDF 功能進行發展之外,為確保後續 燃料佈局之正確性與總負載計算之便利性;本研究亦利用 Microsoft 之 Excel 軟 體進行相關試算工具的建立(圖 11),以避免不必要的人因錯誤而導致分析結果 的不適用。

上述之配置結果,可經由 UDF 指令加以描述後,進而在模擬時達成燃料配置之目的;其程式判別式之邏輯可匯整如下:

$$\begin{cases} Row = \frac{\left(x \ value + \frac{1}{2} fuel \ wide\right)}{pitch} \\ Colume = \frac{\left(z \ value + \frac{1}{2} fuel \ wide\right)}{pitch} \end{cases}$$

其中,

Row與Colume代表燃料束由中心向外計算之行列值;

x value,代表該網格之 x 座標值;

zvalue,代表該網格之 z 座標值;

<u>1</u> fuel wide 代表燃料束的寬度,此項係用來修正此一對稱模型於最中央行

列僅有 1/2 燃料寬度之狀況;

pitch則為各行列燃料束中心間距。

透過上述計算,並以無條件括去法求取其行列值之整數後,即可獲得對應之燃料行列數,進而依序給定該燃料所具有之熱源項。最後,經由上述作業程序之

	А	В	С	D	E	F	G	Н
1	0,0	1	2	3	4	5	6	$X \rightarrow$
2	1	73.4	73.4	73.4	14.8	14.8	14.8	
3	2	73.4	73.4	73.4	14.8	14.8	14.8	
4	3	73.4	73.4	14.8	14.8	14.8		
5	4	14.8	14.8	14.8	14.8	14.8		
6	5	14.8	14.8	14.8	14.8			
7	6	14.8	14.8					
8	Z							
9	\downarrow							

圖 11、核二乾貯系統燃料佈局配置規劃試算表

四、分析模式發展、邊界條件定義與案例選擇

基於以上所蒐集之系統資訊與各項運轉參數,本研究利用 FLUENT 之前處 理程式-GAMBIT 將所獲得之幾何形狀資訊進行幾何模型建立、網格生成與邊界 條件設定等作業,以下將逐一就各階段模式發展成果進行說明。

4.1 模式發展與網格生成

如圖 12 所示,在考量核二乾貯系統係為 1/4 對稱之幾何設計、且其原始設 計中亦採用裝載上限之熱負載來加以涵蓋各式燃料,故其內部對流之邊界條件亦 可視為對稱。因此,本研究採用 1/4 對稱模式之設定來進行分析模型之發展,除 了希望減少計算所需之資源外,亦藉由適度減少不必要的計算域範圍,而將 TSC 內外兩側之對流區間之網格予以加密,以符合本計畫第三、第四季進行空氣流道 之流場分析需求。

而得利於此對稱模型的處置,本研究得以將所有對流區間之網格分佈進行加密;例如,圖 13所不論是空氣流道或是 TSC 內部燃料匣、中子屏蔽板等結區域 與其組合而成之微小結構,仍能在有限的計算資源限制下,將幾何模型詳實描述。

另外,有鑑於本計畫已建立功能強大之燃料配置 UDF,故已無需再藉由人 工逐一對此1/4 模型中的各個燃料束進行流阻、熱負載與軸向功率進行設定;因 此,所有燃料束以不分區方式統整為一個邊界條件,以利後續分析作業。但由於 燃料束仍然存在所謂的有效發熱段與無熱源之部分,故在分析模式中仍將其區格, 以避免交界網格因為計算誤差所產生的不準度。而最後所獲得之整體模型之網格 分佈則如圖 15 所示;其整體節點數目約為 820 萬左右,較以往未考量 TSC 表面 對流分析之需求之平行驗證用模型[13],本研究所採用之分析模型之總網格增加 約 10%左右。





(b)



圖 13、核二乾貯系統幾何示意圖(a)整體護箱之網格分佈(b)TSC 與內部提籃之網 格分佈



圖 14、核二乾貯提籃、中子屏蔽板與燃料束示意圖



圖 15、核二乾貯系統整體網格分佈

4.2 邊界條件設定

有鑑於本計畫各項熱流分析皆是針對正常貯存狀況下之系統熱流狀況進行 研析,僅需就可能出現之參數,如環境溫度、燃料衰變熱、燃料束類型以及燃料 佈局等進行靈敏度測試;故本分析報告依據台電現有 SAR 之分析設定[3],將邊 界條件歸納如下:

- 考慮輻射熱傳機制
- 由於考慮燃料束內部對流效應,故以多孔性材質模擬,並以均質化處理
 進行材料性質設定;
- 假設空氣通道之對流機制已經建立,並假設其環境溫度為 32℃,在參 數靈敏度測試中,則依需求進行調整。
- 底部假設為絕熱,其它邊界考慮輻射與對流效應;
- 假設總熱負載為17KW,並依照靈敏度分析調整敏束燃料之發熱量,其 負載配置與UDF之建立則如第三章所述;
- 日照輻射熱對混凝土護箱之影響以 10 CFR 71 規範為依據;
- 混凝土底座等與場址相接觸之表面假設為絕熱邊界。
 除此之外,對於數值模式的部分,此計劃之各項邊界條件可臚列如下:
- 所有流固之共軛熱傳邊界,假設為速度為0之無滑動邊界條件;
- 所有進口之邊界設定為壓力進口邊界,其溫度係依案例環境溫度進行調整;
- 所出口邊界設定為壓力出口邊界,其溫度係依案例環境溫度進行調整;
- 所有計算域之固體邊界(如 VCC 外表面),係假設為混合邊界,並依實
 際材料狀況填入表面放射率、表面熱對流係數與環境溫度,以模擬對流、
 輻射熱傳等效應。

4.3 靈敏度測試與案例選定

由第二章所載之能量守恆方程式可以得知,對於影響乾貯系統熱流特性之參 數主要可以概分為源項的變化以及移熱能力的變化(表 12)兩個部分;而其中源 項的部分則包括了燃料的運轉與冷卻歷史、裝載的數量與燃料束在提籃內的配置 方式等皆有可能影響貯存時的溫度分佈。反之,影響能量對外散逸之部分,經合 理排除因為系統狀態改變(例如內部填充氣體性質改變)等狀況後,預期將僅有環 境溫度的變化會影響系統元件之溫度值。

為求能簡化未來進一步進行設計壽命期間參數變化之總量,本節將針對上述 各項參數進行靈敏度測試,以期求得各參數之影響程度並獲得足以涵蓋後續分析 需求之參數設定。

表 12 影響設計壽命期間熱流特性之參數

熱散失影響參數	對應模擬方式
燃料類型	採用 3.5 節之技術
燃料冷卻歷史	採用 3.3 節之技術
燃料運轉歷史	採用 3.3 節之技術
燃料佈局	採用 3.4 節之技術
源項影響參數	對應模擬方式

環境溫度 直接調整邊界條件之溫度

4.3.1 環境溫度之影響評估

由於乾貯系統之熱量最終將藉由與空氣之熱對流以及對外界輻射來散逸至 環境之中,因此環境溫度的改變預期將造成:

1·空氣性質的改變

2·輻射熱傳能力因絕對溫度的變化而提高

但由於上述兩個影響之改變量在合理的貯存條件下,其變化程度即十分有限。 也因此,本節之評估工作主要係針對合理參數變動時所造成的影響是否偏離線性 關係進行探討,並藉以作為簡化此參數影響之依據。

考量原有 SAR 中將 42°C 環境溫度之條件視為異常案例進行評估,故若將 合理之貯存環境溫度範圍界定於正常環境溫度的±10°C 之內,或可包含所有可 能之環境溫度變異。

圖 16 係以環溫 32°C 為基準時,計算不同環溫變化下之元件溫度變化,此 結果顯示,各個不同環境溫度變化案例之所有元件的溫度係為線性變化。而由圖 16 中之 R 值(回歸平方和/總平方合)之差異量只有 0.0041 可以證實,在合理的環 境溫度變化區間內,所有乾貯系統元件之溫度因為氣體性質與絕對溫度差異所造 成的影響極為有限。也因此,後續研究中可以忽略環境溫度之影響以利繁瑣分析 工作之簡化與研究方向的聚焦。



圖 16、核二乾貯系統各元件溫度受環境影響之指標

4.3.2 燃料類型之影響評估

如圖 17 與圖 18 與所示,當核二乾貯裝載不同設計之燃料時,其所具有的 不同軸向功率分佈,將可能進一步造成整體溫度之差異;例如,當 TSC 內裝載 GE8X8 燃料時,其提籃頂端具有比裝載 ATRIUM-10 燃料略高的溫度。這是因為 原有 SAR 中採用分段階梯的軸向功率分佈,使得提籃頂端具有較大的功率集中 效應所致,並促使燃料之溫度略高於裝載 ATRIUM-10 之燃料。此外,由於核二 乾貯系統係採用對流式提籃的設計,因此經由燃料底部流入的氦氣,將會在受燃 料加熱後,向上流出燃料束,再透過 TSC 殼體內側之自然對(圖 19),與 TSC 外 側的空氣對流機制達成能量移除之目的。

也因此,燃料束之擺置差異除了造成燃料束與其相鄰之中子屏蔽板與提籃之 溫昇外,亦同時對局部 TSC 溫度分佈造成影響(如圖 20)。這是因為 GE-8X8 燃 料與 ATRIUM-10 燃料之不同軸向功率,導致此系統在裝載 GE-8X8 燃料時,會 有較多的熱量是經由對流攜至 TSC 頂部;而非藉由輻射直接用提籃側邊向外散 逸,並使得 TSC 頂部具有較高之溫度分佈。但隨著 TSC 內部的流動持續受外部 流場冷卻之故,此一受不同燃料所致的 TSC 溫度差異,將會隨 TSC 高度的減低 而逐漸消失。

進一步的量化比對更顯示,在裝載 GE-8X8 燃料的核二乾貯系統除了將具有 較為保守的溫度值 (表 13)外,在考慮 SCC 大多由 TSC 底部開始發生之實際狀 況,本研究仍選用與原安全分析報告中一致之燃料類型,進行後續分析工作。



圖 17、核二乾貯系統裝載 GE8X8 時之系統溫度分佈



圖 18、核二乾貯系統裝載 ATRIUM-10 時之系統溫度分佈



圖 19、核二乾貯系統內部對流現象

元件名稱	GE-8X8	ATRIUM-10
燃料護套	199.76	193.97
BORAL	196.32	193.61
燃料提籃	196.63	193.92
TSC 殼體	135.58	134.78
TSC 上蓋	146.2	141.29
內襯	71.14	70.839
VCC	70.91	70.608

表 13 核二乾貯系統裝載不同燃料時之元件最高溫度(°C)



圖 20、核二乾貯系統貯存不同燃料元件時 TSC 之溫度分佈差異

4.3.3 燃料運轉歷史與佈局之影響評估

經過前一章節之討論後得知,當提籃內燃料束之熱量經由燃料內部之對流機 制與側邊輻射移除之比重有所不同時,將進一步使得整體元件之最高溫度與局部 溫度分佈發生變化;為能進一步釐清其它參數對此現象之影響程度,本節隧針對 不同運轉歷史與佈局對整體影響進行綜合評估。

本節評估主要基於總熱負載不變之前提下,探討核二系統裝載不同運轉及冷 卻歷史之用過核子燃料。為確保此一評估可以涵蓋未來可能之任何狀況,故其最 高單束燃料熱負載係以 500W(冷卻超過 5 年)為評估基準,再搭配其它較低功率 之燃料束來達成佈局之設計;而值得注意的是,為確保裝載燃料之最低熱功率之 合理性,本報告參考核一裝載時之實際最低燃料負載來作為配置組合之依據,而 透過此合理裝載功率之上下限控制,即可獲得可涵蓋所有實際狀況但卻不致過度 保守之分析規劃。

其中,TYPE-I(圖 21a)之配置係以中央集中之規劃來進行設計,以期求得最為保守之系統元件溫度;反之,TYPE-III(圖 21c)之配置即是將最熱之燃料置於最外側,用以探討非集中擺置對元件最高溫度與 TSC 溫度分佈之影響;而具於兩者之間的 TYPE-II 配置(圖 21b)則是作為比對之用。

圖 22 為三種負載之整體溫度分佈;其結果顯示當採用密集擺置時,密集擺 置得確會導致較高的燃料溫度(圖 23),並導致系統中央之元件溫度(圖 24&表 14)較其它配置更高。然而,除了 TYPE-III 之配置外,上述各式配置燃料提籃的 配置對外元件之影響並無顯著影響。這是因為 TSC 內部之熱量需透過 TSC 外殼 的輻射與熱對流向外傳遞,故各式配置對於 TSC 與外界元件的影響將因此而抑 低。反之,當將採用 TYPE-III 之配置而將較熱之燃料置於外側時,提籃中央可 提供自然對流驅動力之熱源即隨之減少。

有趣的是,此一變化除了令提籃外側之溫度升高外,亦同時使得提籃直接對 TSC 外殼輻射熱傳的能力提昇,進而使對流與輻射熱傳的比例發生改變,並造成

軸向溫度(圖 26)趨勢差異如下:

- 在 0.8m 以下 TYPE-III 的溫度值與其它配置皆無明顯差異,這是因為基於能量守恆之故,所有內部之能量需透過 TSC 來傳導,故處於 整個空氣流道最上游(內部對流最下游)的區間,其溫度主要係受進 口條件影響;
- 2. 在 0.8m-3.2m 之間, TYPE-III 具有比其它區間更高的溫度;這是因為直接將較熱燃料置於提籃外側時,將使該燃料的有效發熱段直接使得提籃的側邊溫度提高(圖 27)並透過輻射熱傳使此區間之 TSC 溫度較其它配置更為一致;
- 3. 在 3.2m 以上的區段, TYPE-III 的配置則將具有最低的 TSC 溫度; 此點與上項所述,較多的能量透過提籃側邊傳遞而未經由燃料內部 對流移除,而使得累積於 TSC 頂端的總熱量減少之故。

透過以上之評估可以發現,除了 TYPE-III 之特殊配置外,其它配置對於整 體 TSC 之溫度分佈影響並不明顯;而若更進一步的檢視 SCC 機制之首要發生位 置係座落於 TSC 下半部之處,故在考量元件溫度、SCC 評估之兩項評估之保守 性後,本研究決定以 TYPE-III 與原始 SAR 之平均分佈進行設計壽命期間之熱流 分析設定。

(8	a) _{0,0}	1	2	3	4	5	6	Х→
	1	500	500	500	500	49.2	49.2	
	2	500	500	500	49.2	49.2	49.2	
	3	500	500	500	49.2	49.2		
	4	500	49.2	49.2	49.2	49.2		
	5	49.2	49.2	49.2	49.2			
	6	49.2	49.2					
	Z							
(b)	0,0	1	2	3	4	5	6	X→
	1	500	250	500	250	250	250	
	2	250	225	500	100	100	100	
	3	500	500	100	100	100		
	4	250	100	100	100	100		
	5	250	100	100	100			
	6	250	100					
	Z							
(c)	0,0	1	2	3	4	5	6	Х→
	1	49.2	49.2	49.2	49.2	49.2	500	
	2	49.2	49.2	49.2	49.2	49.2	500	
	3	49.2	49.2	49.2	49.2	500		
	4	49.2	49.2	49.2	49.2	500		
	5	49.2	49.2	500	500			
	6	500	500					
	Z							

圖 21、不同燃料佈局之案例(a)TYPE-I、(a)TYPE-II、(a)TYPE-III。



圖 22、不同燃料佈局案例之整體溫度(°C)分佈(a)TYPE-I、(a)TYPE-II、(a)TYPE-III。



圖 23、不同燃料佈局案例於燃料有效發熱段頂端之橫截面溫度(°C)分佈(a)TYPE-I、(a)TYPE-II、(a)TYPE-III。

	FUEL	BORAL	BASKET	TSC	LINER	VCC
UNIFORM	199.8	196.3	196.6	135.6	71.1	70.9
TYPE-I	234.9	216.5	216.7	140.5	71.3	71.1
TYPE-II	223.5	203.7	203.4	136.5	70.2	69.9
TYPE-III	197.9	172.8	172.5	129.0	69.1	68.8

表 14 核二乾貯系統裝載不同配置時之元件最高溫度(°C)



圖 24、不同燃料配置下之徑向溫度(°C)分佈



圖 25、不同燃料配置下之 TSC 温度(°C)分佈



圖 26、各式配置之 TSC 溫度(°C)趨勢比對



圖 27、各式配置之 TSC 無因次溫度趨勢比對

五、設計壽命期間之熱流特性及空氣流道流場研析

透過以上各個參數之靈敏度分析結果,本章將選擇以裝載 GE-8x8 燃料、具 有均勻與密集分佈之燃料佈局,並在剔除環境溫度之影響後,針對本計畫原訂之 工作規劃進行核二乾貯系統在貯存年限期間之熱流與 SCC 效應之研析、系統設 計壽命期間各元件溫度趨勢預估工具建立以及空氣流道內流場研析與鹽類附著 之可能位置探討。

有鑑於核二機組目前已運轉約 33 年,且未來實際施行乾貯作業仍需若干時 程進行規劃準備之故;本研究於分析之貯存時程,時係由核設施與乾貯系統執照 有效時間之總合再加 20%,以涵括所有可能之貯存狀況;如初期選擇低功率裝載 之熱測試作業所可能出現之極低負載規劃。以下各節將就設計壽命期間之熱流特 性分析結果、SCC 分析結果以及空氣流道內流場之影響所獲得之研究成果逐一 進行說明。

5.1 核二廠乾式貯存設施密封鋼筒設計壽命期間元件溫度分布模擬研析

為進一步探討不同初始燃料配置於設計壽命期間之熱特性,本研究先行利用 第三章所建立之衰變熱計算模式,針對不同初始裝載之燃料進行後續冷卻 100 年之燃料衰變熱估算作業,其所得結果則如圖 28 所示。其結果合理地顯示當燃 料束裝載時之冷卻時間較短時(如 500W/束之燃料)其貯存初期之衰變熱變化將會 遠高於原先 SAR 中所規劃裝載之燃料;同樣地,用以與高發熱量燃料搭配組合 的燃料(如 100W/束之燃料)因為已充分冷卻之故,故其衰變熱變化就較為和緩。

如圖 29 所示,兩種不同配置之燃料在貯存兩年後,其系統內部溫度因為受 到密集擺置效應之影響而具有近30°C之溫差;但若貯存時間持續增加 (圖 30), 則兩種配置之溫度將逐漸縮減至 10°C 左右。這是因為各個貯存時間下,無論是 平均配置或是 TYPE-I 的集中配置的總負載差異並不明顯;但在採用密集配置的 TYPE-I 中,因此裝載了數根 500W 的燃料將因為貯存時間的增加而具有較大的 熱負載變化之故,而使整體溫度趨勢在設計壽命期間,因各別燃料功率衰退比例 的不同而呈現差異。

如同先前所探討過的,燃料提籃內最外圍燃料之熱功率將會是影響 TSC 表 面溫度的影響參數之一,故當分析時採用 TYPE-I 之密集配置時,即意味著小幅 的降低了最外圍燃料之熱負載;此一差異雖然不致於如先前討論過的 TYPE-III 般大幅影響整體之 TSC 溫度分佈,但仍會對 TSC 之系統產生些微之影響。如圖 31 與圖 32 所示,在平均燃料佈局與 TYPE-I 之密集佈局中,可以發現不同貯存 年限之平均配置,其 TSC 之軸向溫度分佈係呈現近乎比例變化之趨勢;反之, 在密集擺置的配置中,即會發現 TSC 之下半部溫度分佈,受到貯存時間增長而 減低的情況較 TSC 上半部來更不明顯。這是因為在 TYPE-I 的密集配置方式,將 將會導致貯存時間增加後,內部需藉由燃料內部對流移熱的高熱源燃料之熱負載 下降,進而使需藉由對流將熱帶至 TSC 頂部的熱源比例減少。

最後,在匯整不同配置下各貯存年限之元件溫度後可以發現以下幾點:

- 由於密集擺置效應之故,TYPE-I之燃料與提籃溫度於各貯存年下皆
 比均勻擺置之結果為高;
- 整體溫度趨勢之變化將與隨著衰變熱的變化成正比,於裝載初期具 有較大的溫降現象;
- 3. 對於長期貯存最需加以考慮之TSC表面溫度,採用密集擺置將可獲 得較大的最高溫度;且經上述之討論後可以確認,主要係增加TSC 上半部之溫度;
- 4. 整體而言,核二乾貯系統於設計壽命期間之熱流現象與核一乾貯相 似,但因為兩套乾貯系統提籃設計與裝載規劃之不同,而使兩者之 最高溫度及 TSC 溫度分佈趨勢有所不同;但經評估相關溫度範圍後 確認發生 SCC 之機率仍以 TSC 底部較高。



圖 28、不同初始裝載之燃料於分析貯存期間之衰變熱變化趨勢



圖 29、貯存2年後之整體溫度分佈(a) 平均配置 (b)TYPE-III 配置



圖 30、貯存 100 年後之整體溫度分佈(a) 平均配置 (b)TYPE-III 配置



圖 31、平均配置下之 TSC 溫度在不同貯存年份下之變化趨勢



圖 32、TYPE-I 配置下之 TSC 溫度在不同貯存年份下之變化趨勢

5.1.1 設計壽命期間元件溫度研析成果匯整

經由本報告第四章對各項參數之測試與評選後,本研究於本節針對保守條件 下之貯存狀態進行了設計壽命期間之元件特性研析,而經前節之討論後所獲成果 可匯整如下:

- 乾貯系統之各項元件溫度係隨貯存時間而呈現指數遞減之趨勢,故在貯 存期間之熱流溫度與作業時限可由初始裝載之功率來加以涵蓋;
- 由於較短冷卻時間的燃料束,貯存後功率衰變趨勢較大,故其冷卻速率 將比均勻擺置更為顯著;
- 3.在設計壽命期間,整體乾貯系統之熱傳機制並無任何改變;亦即在針對 設計壽命、老化等因素進行評估時,皆需將乾貯系統之熱傳機制對其所 在意之參數(如溫度或溼度)納入考慮。
- 4.最後,經過本節之研析與報告撰寫,已依約完成計畫書中所載之相關工作。

元件\貯存年份	0Y	2Y	10Y	20Y	40Y	80Y	100Y
FUEL	199.77	194.32	169.73	148.68	115.05	78.50	64.82
BASKET	196.63	190.02	166.73	145.99	113.30	76.35	61.97
TSC-SHELL	135.58	131.50	116.66	102.52	81.05	67.81	48.43
LINER	71.14	69.56	62.79	57.35	49.20	40.15	37.22
VCC	70.91	69.45	62.62	57.21	49.07	40.07	37.21

表 15 核二乾貯系統均勻擺置時,經不同貯存年份下之元件最高溫度(°C)

表 16 核二乾貯系統於密集擺置時,經不同貯存年份下之元件最高溫度(°C)

元件\貯存年份	0Y	2Y	10Y	20Y	40Y	80Y	100Y
FUEL	234.9	228.8	204.0	169.6	146.7	85.0	69.3
BASKET	216.4	211.2	188.8	155.6	138.4	82.2	68.8
TSC-SHELL	140.5	138.0	123.6	111.6	92.7	57.6	49.7
LINER	71.3	70.0	63.7	55.1	52.8	40.0	37.4
VCC	71.1	69.8	63.6	55.0	52.7	40.0	37.4

5.2 乾貯系統設計壽命期間各元件溫度趨勢預估工具之建立

依據本報告第4章之討論,已充份證實各項參數與系統間之關係如下:

1. 乾貯系統內之燃料衰變熱,可透過ASB 9-2 等類之經驗式加以描述;

2·乾貯系統之各項元件溫度係呈指數變化之趨勢;

3·燃料類型之影響可透過GE8X8進行保守涵蓋;

4·系統所處之環境溫度可以利用線性疊加方式來加以表現。

因此若進一步將 5.1 節中各參數分析所得之結果進行匯整,則可透過上述各 項參數之處置方式,把影響系統元件溫度之因子加以匯整進而建立一套可以快速 將不同貯存時間、初始燃料功率以及環境溫度等參數納入考量,並求取核二乾貯 系統各項元件溫度之預估工具,以供未來管制單位或其它研究工作需要導入合理 熱流參數時使用。

為了達到此一目的,本研究首先針對各項參數之案例於不同貯存時間下所獲 之系統元件以指數函數進行適配法(curve-fitting),而所得結果則如圖 33 所示; 透過指數方程式進行適配將可令此系統之各元件溫度與貯存之時間為:

 $T_{MAX} = 0.9858e^{-0.017Y}$

上述方程式意味著,若能再透過相似之方法,將功率與貯存年限進行適配, 即可將溫度、貯存時間與初始功率等因素加以統整,並獲得不同負載與貯存時間 下的元件溫度。故本研究將已獲得之回歸式與各元件之功率-溫度關係進行處理 後(如圖 35-圖 39 所示),即完成預估工作之所有計算式處理。

最後,本研究以 EXCEL 試算表為基礎,將上述所有回歸式加以統整,並建 立了一套可以快速預估各元件溫度之工具(如附件二)。使用者僅需針對所需之熱 負載以及環境溫度進行輸入,即可求取各個元件在不同時間下的溫度值。

另外,為能更進一步地利用已經建立的熱流預估工具,本研究將進一步地將 TSC 表面之溫度分佈,以其最大溫度與環境溫度進行無因次化(如圖 40 所示)並 作為後續 SCC 預估之基礎,而其詳細發展過程則將於 5.3 節再行說明。

5.2.1 温度趨勢預估工具發展成果匯整

經本節之匯整與評析之發現可匯整如下:

- 1.經由模擬數據及熱流理論之整合,可將各元件之溫度隨熱負載(貯存時間) 之變化分別加以歸納為簡易的線性方程式,而此一結果再與ABS 9-2 衰 變公式整合後,即可作為本溫度趨勢預估工具之核心,以快速提供系統 之元件溫度;
- 此一工具或任何元件之預估式可作為其它領域研究學者之參考依據,以
 提供合理之溫度參數供其定義研究參數之用;
- 3.相較於燃料於設計壽命期間具有近150°C之溫差,其它元件之溫度因其 所在位置,則僅有燃料溫度的1/2(如TSC)或1/5(如VCC)。



圖 33、不同元件經無因次化後的指數適配結果



圖 34、初始裝載功率與貯存年限之關係



圖 35、燃料溫度與功率之關係式



圖 36、提籃溫度與功率之關係式



圖 37、TSC 溫度與功率之關係式



圖 38、內襯溫度與功率之關係式


圖 39、VCC 溫度與功率之關係式



圖 40、TSC 表面溫度與最高溫度之關係式

5.3 核二乾式貯存設施密封鋼筒設計壽命期間之熱流特性與應力 腐蝕關係研析

在經由 5.1 節所獲之核二系統於設計壽命期間之 TSC 溫度後,即可藉由評估 該元件所在之相對溼度與溫度,進一步評估其發生 SCC 之可能性與其裂紋成長 之活化能,進而作為 TSC 表面最先出現 SCC 之可能性評估。

美國電力研究所(Electric Power Research Institute, EPRI)之研究結果指出, 304 系不銹鋼在含有氯鹽溶液之環境中,氯鹽的腐蝕效應將會造成應力腐蝕破裂 之狀況。而其研究結果(圖 42)更指出,不同的鹽類、溫度、相對溼度與絕對溼度 之參數組合皆可能令試片發生 SCC,但此一現象發生的首要條件即是必須於試 片表面存在有溶融的氯鹽溶液。

有鑑於本研究所建立之預估模式可針對任何可能之環境溫度與溼度影響進 行量化分析,而為能先行探討整體趨勢之定性影響,本節即先行就廠址附近之平 均環境溼度(圖 41)與 TSC 之溫度,先行就定性評估部分進行探討,以作為後續 5.4 節之定量評估的依據。基於以上說明,用以合理評估材料是否處於 SCC 狀況 之關鍵參數-相對溼度(Relative Humidity),可由下式計算求得:

$$RH = \frac{\rho_W}{\rho_{W,MAX}};$$

其中,
$$\rho_W = \frac{e}{R_w T}$$
;

e = exp(20.386 - 5132K/T)mmHg;

 $R_w = 461.52 \text{ J/(kg K)} \circ$

計算所得之溼度與溫度條件若係處於 SCC 發生區間時,則需藉由以下公式 針對活化能進行計算:

$$\frac{da}{dt} \begin{cases} \alpha = exp\left[-\frac{Q_g}{R}\left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right] \text{ for } RH \ge DRH \text{ and } K1 > 0\\ \alpha = 0 \qquad \qquad \text{ for } RH < DRH \text{ or } K1 \le 0 \end{cases}$$

其中, Q_q 為目前 EPRI 暫時最保守活化能(40KJ/mol),R為理想氣體常數, T_{ref} 為

參考溫度,其值為 80°C(353.15K),而 DRH 則為上述定義之潮解相對溼度。

然而,在進行量化的評估之前,本研究仍依循物管局 103 年之委託計畫中對 核一廠 TSC 表面相對 SCC 劣化速度的探討方式,先行以無因次化(Normalized) 之方式探討不同環境條件對 SCC 造成之相對裂紋成長率(Relative Crack Growth Rate, R-CGR)來進行定性討論,並作為後續定量評估之依據。

圖 43 為核二系統在不同裝載年份下之,假設環境溼度為 80%下,計算所得 之 TSC 表面相對溼度分佈趨勢;在依循 EPRI 所介定之可能相對溼度範圍加以標 記後可以發現,當護箱裝載之初具有高溫的 TSC 殼體整體來說並不致落於鹽類 潮解之區間內;但在 TSC 底部與承載座相接觸之區域,因受到傳導熱傳而出現 較低溫度,並使該區之相對溼度高於潮解閾值,而可能出現鹽類潮解之狀況。

隨著貯存年份的增加,內部衰變熱的逐漸下降亦有可能使得 TSC 表面相對 溼度符合潮解條件之面積增加;例如,而當貯存年限接近 80 年時,其整體之 TSC 表面之相對溼度,皆已坐落於 15-80%之範圍內。此結果雖然顯示,隨著貯存時 間的增加,TSC 表面可能形成 SCC 之誘發環境;但若更進一步檢視在各時間點 下之活化分佈(圖 44)則可以更進一步的發現,整體 TSC 表面活化能之大小係為 一溫度之正相關函數,故當初期溫度高時,雖然相對活化能較高,但此時整個 TSC 表面也因為高溫而使相對溼度低於 SCC 的發生環境,因此並不會發生 SCC 效應。

圖 45 為進一步將 EPRI 之溫度與溼度作為判斷式進行評估後之有效劣化分佈圖。其分佈係定義為,當溫度大於 24°C 且相對溼度介於 15-80%之區間時,所 有 TSC 表面裂紋之 R-CGR,實際對 TSC 殼體造成之有效劣化(EFF-SCC)狀況。 而此結果更清楚的證實了 TSC 表面的相對溼度與活化能,係互為溫度之正相關 與負相關函數,故需在兩條件皆成立之範圍方可對 TSC 表面造成 SCC 效應,且 此效應的強度更將隨著溫度的降低而逐漸趨緩。此一結果或可解釋何以美國各電 廠在進行相關乾貯作業多年後,就護箱表面進行目視檢測而仍未發現任何密封鋼 筒出現應力腐蝕狀況之現象。



圖 41 台北近年之相對溼度



圖 42 304 不銹鋼於不同溼度與溫度條件下發生 SCC 之分佈範圍



圖 43、不同貯存年份下 TSC 表面之相對溼度



圖 44、不同貯存年份下 TSC 表面相對活化能分佈



圖 45、不同貯存年份下 TSC 發生有效劣化之分佈圖

5.3.1 熱流特性與應力腐蝕關係研析成果匯整

本節之研究成果可以歸納為以下各點,以供管制單位或社會大眾參考:

- 1·SCC之活化能與發生閾值係為溫度之相反函數,亦即當溫度過高時TSC 表面往往因為不符合SCC之發生條件,故無需考慮此狀態下的活化能變 化;
- 2.反之,由於TSC對流入空氣的加熱作業,將會使相對溼度下降,因此因此若TSC表面的潮解溼度符合SCC閾值亦代表其,其活化能強度即被限制,而使SCC發展速度受到限值;
- 3.基於以上所述,TSC 表面最活躍之 SCC 發生位置將隨著乾貯系統的貯存時間增加而由下往上移,這意味著 SCC 的可能發生位置並不會隨貯存時間的增加而集中於特定區域。
- 4.綜合以上之評估,可以確定目前採用試片所得之測試結果,將遠比實際 發生於乾貯系統之 SCC 更為保守。

5.4 乾貯系統空氣流道之流場研析與鹽類附著之可能位置分析

除了上述各章中已完成討論的乾貯設計壽命期間之熱流與 SCC 現象,得以 利用計算流體力學技術加以解析外,對於以往常受忽略的空氣流道部分,在導入 更為精細之分析模式後,更能對其狹小的空氣流道內的熱傳機制進行探討,以瞭 解其內部自然對流所形成的流場特性,並初步地瞭解鹽類傳輸時可能附著的位 置。

如圖 46 所示,在長期貯存時空氣會經由乾貯系統底部的狹長進氣口平穩地 流入,但無是核一或是核二乾貯的設計,為避免外來物直接進入到內部垂直之空 氣通道外,故除了在進口處以格網或是其它結構件加以阻擋外,在流道轉折處往 往設計(圖 47 紅色圈)了一個具有轉折流道的底座焊件。此一設計使得流體在流 經垂直流道時,將無法直接向上,而需向前流動至底座焊件底部,再反向流動後 才能進入垂直流道(如圖 46)。

另外,如圖 48 所示之 A-A 截面處,核二乾貯系統在兩個相鄰的底部進口末端,設計了一個分隔區(鞍部)以作為結構強化之用。此一鞍部於流場中就如同一個背向階梯,故當流體開始向上流動時,將會在鞍部(圖 48)、底座焊件的底部以及 VCC 內襯的末端(圖 49)處形成明顯的回流區。上述之幾何設計,形同在原本平順的路道路增設了許多阻礙物,因此若被空氣挾帶進入系統的物質(如液滴或細砂)將會因為慣性力之故而無法快速的改變行動路徑,而被拘限於該處。

如圖 50 所示,當流體進入垂直空氣流道後,其速度分佈雖然逐漸的趨於穩 定,但仍會隨著高度而有若干的變化。例如,在 0-1.5M 之間,雖然 TSC 殼側具 有比 VCC 內襯更高的溫度,但空氣流道的主要速度分佈反而是在靠 VCC 內襯 側具有較高的流速。這是因為雖然表面溫度與流體之溫差將是造成自然對流的主 因,但由於在垂直進口的區域受到上游底座焊件流道彎蜒之影響,而使得流入垂 直流道的氣體具有朝向內襯側的額外速度分量,並造成在此區域內流場並不貼附 在 TSC 表面的現象。然而,此一現象雖然預期將使得該區域的 TSC 表面移熱能

力較差,但也因為逐漸受熱的空氣與內部被挾帶的微粒(液滴或砂土)將會有一段 極長的距離皆與 VCC 內襯作用。故可以預期的,若有部分液滴因挾帶而進入垂 直流道,將因為流場的作用與內襯的加熱效果而沉積於內襯側,而間接的減少了 TSC 底部殼體表面的可能鹽類沉積量。

隨著流體受到 TSC 的逐漸加熱與高度(1.5~2.5M)的逐漸昇高, TSC 表面將逐 漸的對空氣流道造成較大的浮力貢獻, 也因此在此一區間內的最高流道速度, 將 逐漸的由內襯側偏移至 TSC 殼側, 而類似的情形在更高的 2.5-4.0M 處亦將隨著 高度的增加而更形顯著。雖然此時的 TSC 表面速度分佈與速度梯度將比 VCC 內 襯側更高, 但因為高度與溫度的增加, 因此能被挾帶至此的液滴將比先前更為細 小, 甚至可能因為熱空氣的作用而早已乾化而無法藉由液體的表面張力貼附在 TSC 表面。

為能證實上述由流場分佈所獲得的可能鹽驗累積位置之推測,本研究更進一 步地利用雙流體模式,模擬不同粒徑的液滴在空氣流道內被挾帶的現象,而獲結 果則如圖 51 與圖 52 所示,當液滴粒徑約略等於 1mm 時乾貯系統內自然對流的 流速(~0.5m)並不足以挾帶如此大的液滴;因此所有的流體正如先前對速度場的 討論般皆積存於流道的轉折處與最低點,而難以挾帶至 TSC 表面。

而當粒徑縮小至 0.1mm 時則可發現部分的液滴可以被挾帶至垂直流道的表面,但因為流場分佈之故,絕大多數的液滴將積存於水平流道、VCC 內襯側, 而只有少部分有機會被挾帶至 TSC 表面,且因為一開始流體慣性之故,在 TSC 表面將不會有足夠的速度場將液滴挾帶至 TSC 之底部殼體。

最後,若將液滴尺寸縮小至 0.01-0.001mm,則可以很明顯地看到,除了局部因為流體慣性而有較多的流體積存之外,大多數的液滴皆可透過自然對流而挾帶而進入空氣流道,但因為此時液滴之重量將僅有1.3×10⁻⁸~×10⁻¹¹克,因此只需要 2.85×10⁻⁵~×10⁻⁸焦耳的熱量即可將其蒸乾。

因此綜合以上流空氣流道流場特性與液滴之體積關係,可以推定在若鹽類擬透過液滴之方式挾帶進入乾貯系統中,則在過大的粒徑之下大多數的液低將會積

存於底部空氣流道進口的水平段;而略小的液滴可能可以透過挾帶的方式再更深 入空氣流道中,但因為幾何構造所致的流場特性,液滴將會因為慣性力而附著於 底座焊件的鞍部以及 VCC 內襯的表面,最後若空氣中含有極為細小的液滴粒子 且可被空氣流道的自然對流所挾帶,則此時該液滴恐因為體積過小而流入不久後 即被 VCC 內襯與 TSC 表面所散發的熱量所蒸乾。因此對於 TSC 表面的鹽類附 著或沉積速率,將合理地低於以裸露試片所蒐集得到之沉積速率。

5.4.1 流場研析與鹽類附著可能位置之研析成果匯整

基於以上利用液滴於低流速流場中的挾帶現象與討論,可歸納出以下數點供 管制單位與社會大眾參考:

- 1.受限於乾貯系統中自然對流速度有限之故,約略大於 1mm 的液滴將難以被挾 帶至空氣流道中,並使其所含鹽份沉積於 TSC 表面;
- 2.由於乾貯系統之進氣口與底部蜿蜒流道的設計具有分離液滴的能力,因此愈 大的液滴愈容易被此一流道設計局限於進氣口段;
- 3.過小的液滴只需極少的熱量即會被蒸發,故能被自然對流所挾帶的液滴往往 在上昇至 TSC 表面時即已乾化,故無法貼附於 TSC 表面;
- 4.上述發現或可解釋為何目前美國貯存近20年之乾貯系統,其TSC表面含鹽 量遠低於露天測試結果。



圖 46、核二乾貯系統空氣流道內溫度及流場分佈



圖 47、核二乾貯系統底座焊件工程圖



圖 48、核二乾貯系統底座附近流場分佈圖



圖 49、核二乾貯系統底座與空氣流道速度分佈



圖 50、整體空氣流道速度分佈



圖 51、不同粒徑在流道內的分佈差異



圖 52、核二乾貯系統空氣流道底部流體蓄積現象

六、結論

本計畫已依原計畫書所載之時程規劃完成所有研究項目,其中包括了核二乾 貯分析模式之建立、參數靈敏度測試、貯存年限之熱流分析與相應之 SCC 分析、 預估工具發展以及流道內流場研析與鹽類可能沉積位置預估。本研究在完成相關 分析技術之建立後,本研究更利用所發展之分析模式與技術進行參數靈敏度分析, 以期針對各式影響熱流特性之參數;如:環境溫度、系統熱負載、燃料類型等。 其結果指出,在合理的正常貯存條件下環境溫度之影響可以線性修正之方式來加 以評估而無需將考量溫度變化造成的空氣性質差異納入考量。此外,燃料類型之 影響亦證實原有 SAR 所採用之燃料類型確可提供較為保守之分析結果。經由上 述之靈敏度分析後,本研究選擇以 GE-8X8 燃料類型並搭配最高 500W 之單束熱 負載於 17KW 之總熱負載下,以平均分佈及集中擺置進行不同貯存年分下之熱 流分析與 TSC 表面 SCC 特性之分析。

上述設計壽命期間之分析結果顯示,不同裝載配置之核二乾貯系統,乾貯系 統之元件溫度將會隨貯存時間的增長而逐步減低,並使 TSC 表面之空氣的相對 溼度逐漸上昇,而達到可能發生 SCC 之環境溼度;但幸運的,整體綜合分析顯 示空氣流道內的溼度與活化能係互為溫度之倒數關係,故裝載初期 TSC 雖然具 有高溫造成的高活化能,但因為相對溼度過低而不致發生 SCC;反之,當裝載 時間大幅增長時,相對溼度雖然隨著熱負載的降低而提高,但隨之下降的活化能 亦使 TSC 表面之裂紋活化能明顯抑低。整體而言,TSC 表面發生 SCC 之可能性 與嚴重性應不致於像實驗試片般輕易地發生 SCC 裂穿之現象。

最後,本研究藉由 CFD 模擬來探討空氣中含鹽液滴在空氣流道內的挾帶現象,其結果顯示受限於乾貯系統內自然對流流速度有限之故,過大的液滴將無法 順利地被挾帶到 TSC 鋼筒表面;大多被 VCC 底部用以支撐 TSC 的彎曲流道加

以阻擋;反之,而過小的液滴因體積微小之故,只需極小的能量即可將其蒸乾而 令其失去附著於 TSC 表面的機制,故可合理推測在實際乾貯系統之表面鹽類沉 積速度,將遠小於一般試片之測試結果。

參考文獻

- 行政院原子能委員會,"核能電廠用過燃料池貯存格架改裝安全分析報告審查 規範,"民國 79 年 2 月 29 日, (79) 會核字第 1710 號。
- 2台灣電力公司,"核一乾貯系統安全分析報告,"民國 97 年 11 月。
- 3 台灣電力公司,"核二廠乾貯系統安全分析報告,"民國 100 年 12 月。
- 4 施純寬、曾永信"乾式貯存設施設計壽命期間熱傳行為分析期末報告,"民國 103 年 12 月。
- 5 ANSYS Inc., FLUENT 12.0 User's Manuals, 2009.
- 6 NRC, "Impact of Variation in Environmental Conditions on the Thermal Performance of Dry Storage Casks," U.S. NRC , NUREG-2174, 2015.
- 7 施純寬、曾永信,"假設複合式災害情節對乾式貯存設施營運影響分析期末報告,"民國 100 年 12 月。
- 8 施純寬、曾永信,"核二廠用過核子燃料乾式貯存設施熱傳安全驗證研究期末 報告,"民國 101 年 12 月。
- 9 施純寬、曾永信,"乾式貯存設施於假設複合式災害事故下之熱流分析暨密封 鋼筒內部對流監測機制研究期末報告,"民國 102 年 12 月。
- 10 施純寬、曾永信,"乾式貯存設施設計壽命期間熱傳行為分析,"民國 103 年12 月。
- 11 台電/核能研究所,"計算流體力學分析方法論/乾式貯存與大修機組冷卻之熱 流分析應用,"民國 99 年 12 月, ITITRAM CS/KS-CFD-MHD-01。
- 12 ANSYS Inc., "FLUENT UDF Manual," 2009.
- 13 施純寬、曾永信,"核二廠用過核子燃料乾式貯存設施熱傳安全驗證研究,"民國 101 年 12 月。

附錄一

FLUENT 燃料束源項程式碼

```
/*
       Viscous Resistance
                                 t)+C_V(c, t)*C_V(c, t)+C_W(c, t)
                                                                  Diameter*/
Profile UDF in a Porous Zone
                                 t)*C_W(c, t));
that utilizes F_PROFILE*/
                                     F_PROFILE(c, t, i)
                                                           =
                                0.1015*pow(a*C_R(c,t)*Dh/C_
                                                                    begin_c_loop(c,t)
#include "udf.h"
                                MU_L(c,t),0.8)/Dh/Dh*1.7812
                                                                    {
                                 096;
                                                                      F_PROFILE(c,t,i)
                                                                                             =
                                                                  32./Dh/Dh/1.7812096;
DEFINE_PROFILE(friction_fac
                                  }
tor_ATRIUM_10,t,i)
                                   end_c_loop(c,t)
                                                                    }
                                                                    end_c_loop(c,t)
{
  real x[ND_ND];
  real a,Dh;
                                 }
                                                                  }
  cell_t c;
Dh=12.3225E-3; /* Hydrolic
                                DEFINE_PROFILE(friction_fac
Diameter*/
                                                                  DEFINE_PROFILE(K_LTP_C, t, i)
                                 tor_LAM,t,i)
                                 {
                                                                  {
                                   real x[ND_ND];
                                                                    real x[ND_ND];
                                   real a,Dh;
 begin_c_loop(c,t)
                                                                    real a,Dh;
  {
                                   cell_t c;
                                                                    cell_t c;
    a=sqrt(C_U(c, t)*C_U(c,
                                Dh=12.3225E-3; /* Hydrolic
Dh=12.3225E-3; /* Hydrolic
                                   end_c_loop(c,t)
                                                                  t)+C_V(c, t)*C_V(c, t)+C_W(c, t)
Diameter*/
                                                                  t)*C_W(c, t));
                                                                      F_PROFILE(c,t,i)
                                 }
                                                                                             =
                                 DEFINE_PROFILE(K_LTP_P,t,i)
                                                                  (156.9+168.1*pow(a*C_R(c,t)
                                                                  *Dh/C_MU_L(c,t),-0.546))/2.
 begin_c_loop(c,t)
                                 {
                                   real x[ND_ND];
                                                                  301875*1.7812096*1.7812096;
  {
                                   real a,Dh;
    a=sqrt(C_U(c, t)*C_U(c,
                                                                    }
t)+C_V(c, t)*C_V(c, t)+C_W(c, t)
                                                                    end_c_loop(c,t)
                                   cell_t c;
t)*C_W(c, t));
                                Dh=12.3225E-3; /* Hydrolic
                                                                  }
   F_PROFILE(c,t,i)
                                Diameter*/
                           =
(21.47+168.1*pow(a*C_R(c,t)
*Dh/C_MU_L(c,t),-0.546))/2.
                                   begin_c_loop(c,t)
                                                                  DEFINE_PROFILE(K_FUEL,t,i)
301875*1.7812096*1.7812096:
                                   {
                                                                  {
  }
                                     a=sqrt(C_U(c, t)*C_U(c,
                                                                    real x[ND_ND];
```

real a,Dh,Re; } source = MaxPower*f; cell_t c; end_c_loop(c,t) return source; Dh=12.3225E-3; /* Hydrolic **#if** !PR HOST } Diameter*/ begin_c_loop(c,t) DEFINE_SOURCE(suorce_KSDSS_ if (A> height) { OY, c, t, dS, eqn) { Re=sqrt(C_U(c, t)*C_U(c, { $t)+C_V(c, t)*C_V(c, t)+C_W(c, t)$ real x[ND_ND]; f=1-(A-height)/L1; t)*C_W(c, real A, B, L1, L2, f, p0, } t))*C_R(c,t)*Dh/C_MU_L(c,t) source, height, MaxPower; else if (height>B) $C_{CENTROID(x, c, t)};$: { F_PROFILE(c,t,i) height= x[1]; = ((0.620*4+0.4646*2)+(108.6*)p0=0.66045; f=1-(height-B)/L2;pow(Re, -0.58)*4+2.196*pow(R L1 = 0.15*3.6579;} e, -0.084)+235.7*pow(Re, -0.6 L2 = 3.6579*0.8;else i f 86)*2))/4.362577*1.7812*1.7 A = p0+L1;(height>4.31835) 812096; B = p0+L2;{ } f=0; end_c_loop(c,t) } } MaxPower=199/0.0657400 else 8460044/0.825; f=1; DEFINE_PROFILE(K_UTP, t, i) if (A> height) source = MaxPower*f; { return source: { real x[ND_ND]; #endif /* !PR_HOST*/ real a,Dh,Re; f=1-(A-height)/L1; #if !RP_NODE cell_t c; } Dh=12.3225E-3; /* Hydrolic else if (height>B) if (A> height) Diameter*/ { { begin_c_loop(c,t) { f=1-(height-B)/L2;f=1-(A-height)/L1; $Re=sqrt(C_U(c, t)*C_U(c, t))$ } } $t)+C_V(c, t)*C_V(c, t)+C_W(c, t)$ i f else if (height>B) else t)*C_W(c, (height>4.31835) { t))*C_R(c,t)*Dh/C_MU_L(c,t) { : f=0; f=1-(height-B)/L2; $F_PROFILE(c, t, i) =$ } } (-9.68+9.526*pow(Re, 0.0046) else if else)/0.64389*1.7812*1.7812096; (height>4.31835) f=1;

```
{
                            source, height, MaxPower;
                                                            else if (height>B)
                                 C_CENTROID(x, c, t);
              f=0;
                                                              {
         }
                                 height= x[1];
    else
                                 p0=0.66045;
                                                              f=1-(height-B)/L2;
                                     L1 = 0.15*3.6579;
         f=1;
                                                             }
                                     L2 = 3.6579*0.8;
    source = MaxPower*f;
                                                                 else
                                                                               if
                                 A = p0+L1;
                                                         (height>4.31835)
    return source;
#endif /* !RP_NODE*/
                                      B = p0+L2;
                                                                   {
                                                                      f=0;
                                                                   }
#if !PARALLEL
                                 MaxPower=152.5287/0.06
                                                              else
                            574008460044/0.825;
                                                                 f=1;
    if (A> height)
                                 if (A> height)
                                                              source = MaxPower*f;
                                                              return source;
    {
                                 {
                                                         #endif /* !PR_HOST*/
    f=1-(A-height)/L1;
                                f=1-(A-height)/L1;
                                                         #if !RP_NODE
    }
                                 }
    else if (height>B)
                                 else if (height>B)
                                                             if (A> height)
    {
                                 {
                                                              {
    f=1-(height-B)/L2;
                               f=1-(height-B)/L2;
                                                             f=1-(A-height)/L1;
    }
                                 }
                                                              }
                                                              else if (height>B)
         else
                     if
                                      else
                                                   if
(height>4.31835)
                          (height>4.31835)
                                                              {
         {
                                      {
             f=0;
                                         f=0;
                                                              f=1-(height-B)/L2;
         }
                                      }
                                                              }
    else
                                 else
                                                                   else
                                                                               if
         f=1;
                                      f=1;
                                                         (height>4.31835)
    source = MaxPower*f;
                                 source = MaxPower*f;
                                                                   {
                                                                      f=0;
                                 return source;
    return source;
#endif /* !PARRLLEL*/
                           #if !PR_HOST
                                                                   }
}
                                                              else
DEFINE_SOURCE(suorce_KSDSS_
                                if (A> height)
                                                                  f=1;
10Y, c, t, dS, eqn)
                                 {
                                                              source = MaxPower*f;
{
                                                              return source;
                                                        #endif /* !RP_NODE*/
    real x[ND_ND];
                                f=1-(A-height)/L1;
    real A, B, L1, L2, f, p0,
                                 }
```

```
#if !PARALLEL
                                  MaxPower=120.1149/0.06
                                                                else
                                                                      f=1;
                              574008460044/0.825;
    if (A> height)
                                   if (A> height)
                                                                source = MaxPower*f;
    {
                                   {
                                                                 return source;
                                                            #endif /* !PR HOST*/
    f=l-(A-height)/L1;
                                 f=1-(A-height)/L1;
                                                            #if !RP_NODE
     }
                                   }
    else if (height>B)
                                  else if (height>B)
                                                                if (A> height)
                                                                 {
    {
                                   {
    f=1-(height-B)/L2;
                                 f=1-(height-B)/L2;
                                                                f=1-(A-height)/L1;
    }
                                   }
                                                                 }
         else
                      if
                                       else if
                                                                 else if (height>B)
(height>4.31835)
                             (height>4.31835)
                                                                 {
         {
                                       {
                                                                 f=1-(height-B)/L2;
               f=0;
                                             f=0;
         }
                                        }
                                                                 }
    else
                                  else
                                                                     else if
         f=1;
                                      f=1;
                                                            (height>4.31835)
    source = MaxPower*f;
                                  source = MaxPower*f;
                                                                      {
                                                                           f=0;
    return source;
                                   return source;
#endif /* !PARRLLEL*/
                            #if !PR HOST
                                                                      }
                                                                 else
}
DEFINE_SOURCE(suorce_KSDSS_
                                 if (A> height)
                                                                      f=1;
20Y, c, t, dS, eqn)
                                                                 source = MaxPower*f;
                                   {
{
                                                                 return source;
                                                            #endif /* !RP_NODE*/
    real x[ND_ND];
                                  f=1-(A-height)/L1;
    real A, B, L1, L2, f, p0,
                                   }
source, height, MaxPower;
                                  else if (height>B)
    C_{CENTROID(x, c, t)};
                                                            #if !PARALLEL
                                   {
    height= x[1];
    p0=0.66045;
                                 f=1-(height-B)/L2;
                                                                if (A> height)
         L1 = 0.15*3.6579;
                                 }
                                                                 {
         L2 = 3.6579*0.8;
                                     else
                                                     i f
                            (height>4.31835)
    A = p0+L1;
                                                                 f=1-(A-height)/L1;
         B = p0+L2;
                                       {
                                                                 }
                                                                 else if (height>B)
                                             f=0;
```

}

```
{
                                 {
                                                               {
    f=1-(height-B)/L2; f=1-(height-B)/L2;
                                                              f=1-(A-height)/L1;
    }
                                 }
                                                               }
         else
                     i f
                                 else
                                                  if
                                                              else if (height>B)
(height>4.31835)
                           (height>4.31835)
                                                               {
        {
                                    {
              f=0;
                                           f=0;
                                                              f=1-(height-B)/L2;
         }
                                      }
                                                               }
    else
                                 else
                                                                   else
                                                                               i f
        f=1;
                                    f=1;
                                                          (height>4.31835)
    source = MaxPower*f;
                                 source = MaxPower*f;
                                                                  {
                                                                        f=0;
    return source;
                                 return source;
#endif /* !PARRLLEL*/
                           #if !PR_HOST
                                                                   }
                                                              else
}
DEFINE_SOURCE(suorce_KSDSS_
                                 if (A> height)
                                                                  f=1;
50Y, c, t, dS, eqn)
                                 {
                                                              source = MaxPower*f;
{
                                                               return source;
    real x[ND_ND];
                                 f=1-(A-height)/L1;
                                                         #endif /* !RP_NODE*/
    real A, B, L1, L2, f, p0,
                                  }
source, height, MaxPower;
                                 else if (height>B)
    C\_CENTROID(x, c, t);
                                                         #if !PARALLEL
                                 {
    height= x[1];
    p0=0.66045;
                                 f=1-(height-B)/L2;
                                                             if (A> height)
         L1 = 0.15*3.6579;
                                 }
                                                               {
        L2 = 3.6579 * 0.8;
                                    else
                                                  if
    A = p0+L1;
                            (height>4.31835)
                                                               f=l-(A-height)/L1;
         B = p0+L2;
                                      {
                                                               }
                                           f=0;
                                                              else if (height>B)
                                      }
                                                               {
    MaxPower=58.62069/0.06
                                 else
574008460044/0.825;
                                     f=1;
                                                              f=1-(height-B)/L2;
    if (A> height)
                                 source = MaxPower*f;
                                                              }
    {
                                 return source;
                                                                   else if
                             #endif /* !PR_HOST*/
                                                         (height>4.31835)
                            #if !RP_NODE
    f=1-(A-height)/L1;
                                                                   {
                                                                        f=0;
    }
                           if (A> height)
    else if (height>B)
                                                                   }
```

```
else
                                 else
                                                                   else
                                                                               if
        f=1;
                                     f=1;
                                                          (height>4.31835)
    source = MaxPower*f;
                                 source = MaxPower*f;
                                                                   {
                                                                       f=0;
    return source;
                                 return source;
#endif /* !PARRLLEL*/
                          #if !PR_HOST
                                                                   }
}
                                                               else
DEFINE_SOURCE(suorce_KSDSS_
                                if (A> height)
                                                                  f=1;
100Y, c, t, dS, eqn)
                                                               source = MaxPower*f;
                                  {
                                                               return source;
{
    real x[ND_ND];
                                f=1-(A-height)/L1;
                                                          #endif /* !RP_NODE*/
    real A, B, L1, L2, f, p0,
                                 }
source, height, MaxPower;
                                 else if (height>B)
    C_{CENTROID(x, c, t)};
                                                          #if !PARALLEL
                                  {
    height= x[1];
    p0=0.66045;
                                f=1-(height-B)/L2;
                                                              if (A> height)
        L1 = 0.15*3.6579;
                                  }
                                                               {
        L2 = 3.6579*0.8;
                                                   if
                                      else
    A = p0+L1;
                             (height>4.31835)
                                                               f=1-(A-height)/L1;
         B = p0+L2;
                                      {
                                                               }
                                          f=0;
                                                               else if (height>B)
                                      }
                                                               {
    MaxPower=17.70115/0.06
                                 else
574008460044/0.825;
                                      f=1;
                                                               f=1-(height-B)/L2;
    if (A> height)
                                 source = MaxPower*f;
                                                               }
    {
                                 return source;
                                                                   else if
                            #endif /* !PR_HOST*/
                                                      (height>4.31835)
                            #if !RP_NODE
                                                                   {
    f=1-(A-height)/L1;
                                                                        f=0;
    }
    else if (height>B)
                                if (A> height)
                                                                  }
    {
                                  {
                                                               else
                                                                  f=1;
    f=1-(height-B)/L2;
                                f=1-(A-height)/L1;
                                                               source = MaxPower*f;
    }
                                  }
                                                               return source;
         else
                  if
                                 else if (height>B)
                                                          #endif /* !PARRLLEL*/
(height>4.31835)
                                  {
                                                          }
         {
              f=0;
                                  f=1-(height-B)/L2;
                                                          DEFINE_SOURCE(suorce_GE8X8_
                                                          0Y, c, t, dS, eqn)
         }
                                  }
```

```
87
```

```
{
                                                                 f=1-(A-height)/L1;
    int row, columes;
                                  pitch=0.156624;
                                                                 }
     real x[ND_ND];
                                   fue1_w=0.13406/2.0;
                                                                 else if (height>B)
     real Fxz[6][6]=
                                   axis_x=
                                                                 {
                              (x[0]+fuel_w)/pitch;
     {
          /*Z軸方向→*/
                                   axis z=
                                                                 f=1-(height-B)/L2;
    /*
                              (x[2]+fuel_w)/pitch;
                                                                 }
*/500.,500.,500.,100.,100.,
                                                                      else
                                                                                  if
                                   row=axis_z;
100.,
                                   columes=axis_x;
                                                            (height>4.31835)
  /*
                       Х
                                   fuel_map=Fxz[row][colu
                                                                      {
*/500.,500.,500.,100.,100.,
                             mes];
                                                                           f=0;
100.,
                                                                      }
   /*
                       軸
                                                                 else
*/500.,500.,100.,100.,100.,
                                  if (A> height)
                                                                      f=1;
0,
                                   {
                                                                 source
                                                                                     =
   /*
                       方
                                                            MaxPower*f*fuel_map;
*/100.,100.,100.,100.,100.,
                                   f=1-(A-height)/L1;
                                                                return source;
0,
                                   }
                                                            #endif /* !PR_HOST*/
   /*
                       向
                                   else if (height>B)
                                                            #if !RP_NODE
*/100.,100.,100.,100.,0
                                   {
0,
                                                                 if (A> height)
   /*↓*/100.,100.,0
                       ,0
                                   f=1-(height-B)/L2;
                                                                 {
  ,0 ,0 ,
                                   }
                                       else
                                                    i f
                                                                 f=1-(A-height)/L1;
    };
                              (height>4.31835)
                                                                 }
    real A, B, L1, L2, f, p0,
                                                                 else if (height>B)
                                       {
source, height, MaxPower,
                                             f=0;
                                                                 {
fuel_w, pitch, fuel_map,
                                       }
axis_x, axis_z;
                                  else
                                                                 f=1-(height-B)/L2;
    C_{CENTROID(x, c, t)};
                                      f=1;
                                                                 }
    height= x[1];
                                                                      else
                                                                                  if
                                   source
                                                      =
     p0=0.66045;
                              MaxPower*f*fuel_map;
                                                            (height>4.31835)
    MaxPower=1.0/0.0657400
                                                                      {
                                  return source;
8460044/0.825;
                             #if !PR_HOST
                                                                           f=0;
         L1 = 0.15*3.6579;
                                                                      }
         L2 = 3.6579*0.8;
                                  if (A> height)
                                                                 else
         A = p0+L1;
                                                                    f=1;
                                   {
         B = p0+L2;
                                                                 source
                                                                                    =
```

```
/*
MaxPower*f*fuel_map;
                                                          (x[2]+fuel_w)/pitch;
                              */476.,476.,476.,95.4,95.4,
    return source;
                                                                 row=axis_z;
#endif /* !RP_NODE*/
                              95.4.
                                                                 columes=axis_x;
                                 /*
                                                     Х
                                                                 fuel_map=Fxz[row][colu
                              */476.,476.,476.,95.4,95.4,
                                                            mes];
#if !PARALLEL
                              95.4.
                                 /*
                                                      軸
    if (A> height)
                              */476.,476.,95.4,95.4,95.4,
                                                                 if (A> height)
    {
                              0,
                                                                 {
                                 /*
                                                     方
    f=1-(A-height)/L1;
                              */95.4,95.4,95.4,95.4,95.4,
                                                                 f=1-(A-height)/L1;
                              0,
    }
                                                                 }
                               /*
                                                                 else if (height>B)
    else if (height>B)
                                                      向
                              */95.4,95.4,95.4,95.4,0
     {
                                                                 {
                                                     .
                              0,
    f=1-(height-B)/L2;
                                 /*↓*/95.4,95.4,0 ,0
                                                                 f=1-(height-B)/L2;
                                ,0 ,0 ,
    }
                                                                 }
         else
                       i f
                                  };
                                                                      else
                                                                                   i f
(height>4.31835)
                                                            (height>4.31835)
         {
                                 real A, B, L1, L2, f, p0,
                                                                      {
              f=0;
                              source, height, MaxPower,
                                                                         f=0;
                              fuel_w, pitch, fuel_map,
         }
                                                                      }
    else
                              axis_x, axis_z;
                                                                 else
                                  C\_CENTROID(x, c, t);
                                                                    f=1;
         f=1;
    source
                        =
                                  height= x[1];
                                                                 source
                                                                                     =
MaxPower*f*fuel_map;
                                  p0=0.66045;
                                                            MaxPower*f*fuel_map;
    return source;
                                  MaxPower=1.0/0.0657400
                                                                 return source;
#endif /* !PARRLLEL*/
                              8460044/0.825;
                                                            #if !PR HOST
}
                                       L1 = 0.15*3.6579;
                                       L2 = 3.6579*0.8;
                                                                 if (A> height)
DEFINE_SOURCE(suorce_GE8X8_
                                       A = p0+L1;
                                                                 {
2Y, c, t, dS, eqn)
                                        B = p0+L2;
                                                                 f=1-(A-height)/L1;
{
    int row, columes;
                                  pitch=0.156624;
                                                                 }
    real x[ND_ND];
                                   fuel_w=0.13406/2.0;
                                                                 else if (height>B)
                                   axis_x=
    real Fxz[6][6]=
                                                                 {
                             (x[0]+fuel_w)/pitch;
    {
          /*Ζ軸方向→*/
                                                                 f=1-(height-B)/L2;
                                   axis_z=
```

```
/*
    }
                                                               */392.,392.,78.815,78.815,7
          else
                        i f
                                    if (A> height)
(height>4.31835)
                                                               8.815,0 ,
                                     {
                                                                    /*
          {
               f=0;
                                     f=1-(A-height)/L1;
                                                               */78.815,78.815,78.815,78.8
          }
                                                               15.78.815.0 .
                                     }
                                                                   /*
     else
                                     else if (height>B)
          f=1;
                                                               */78.815,78.815,78.815,78.8
                                     {
     source
                                                               15,0 ,0 ,
                         =
MaxPower*f*fuel_map;
                                     f=1-(height-B)/L2;
                                                                   /*↓*/78.815,78.815,0
    return source;
                                                                ,0 ,0 ,0 ,
                                     }
#endif /* !PR_HOST*/
                                                        if
                                          else
                                                                    };
#if !RP_NODE
                               (height>4.31835)
                                                                    real A, B, L1, L2, f, p0,
                                          {
                                                               source, height, MaxPower,
     if (A> height)
                                               f=0;
                                                               fuel_w, pitch, fuel_map,
     {
                                          }
                                    else
                                                               axis_x, axis_z;
                                                                    C\_CENTROID(x, c, t);
     f=1-(A-height)/L1;
                                         f=1;
                                    source
                                                                    height= x[1];
     }
                                                         =
     else if (height>B)
                               MaxPower*f*fuel_map;
                                                                    p0=0.66045;
                                                                    MaxPower=1.0/0.0657400
     {
                                    return source;
                               #endif /* !PARRLLEL*/
                                                               8460044/0.825;
     f=1-(height-B)/L2;
                                                                         L1 = 0.15*3.6579;
                               }
                                                                         L2 = 3.6579*0.8;
     }
          else
                         if
                               DEFINE_SOURCE(suorce_GE8X8_
                                                                         A = p0+L1;
(height>4.31835)
                               10Y, c, t, dS, eqn)
                                                                         B = p0+L2;
          {
                               {
               f=0;
                                    int row, columes;
                                                                    pitch=0.156624;
          }
                                    real x[ND_ND];
                                                                    fuel_w=0.13406/2.0;
     else
                                     real Fxz[6][6]=
                                                                    axis_x=
         f=1;
                                                               (x[0]+fuel_w)/pitch;
                                     {
                                          /*Ζ軸方向→*/
                                                                    axis_z=
     source
                         =
MaxPower*f*fuel_map;
                                     /*
                                                               (x[2]+fuel_w)/pitch;
                               */392.,392.,392.,78.815,78.
     return source;
                                                                    row=axis_z;
#endif /* !RP_NODE*/
                               815,78.815,
                                                                    columes=axis_x;
                                    /*
                                                        Х
                                                                    fuel_map=Fxz[row][colu
                               */392.,392.,392.,78.815,78.
                                                               mes];
#if !PARALLEL
                               815,78.815,
```

軸

方

向

```
else
                                                            else if (height>B)
    if (A> height)
                                  f=1;
                                                             {
    {
                                source
                                                   =
                            MaxPower*f*fuel_map;
                                                            f=1-(height-B)/L2;
    f=1-(A-height)/L1;
                                return source;
                                                            }
                            #endif /* !PR_HOST*/
                                                                 else
                                                                            if
    }
    else if (height>B)
                            #if !RP_NODE
                                                        (height>4.31835)
                                                                 {
    {
                                if (A> height)
                                                                    f=0;
    f=1-(height-B)/L2;
                                                                 }
                                 {
                                                            else
    }
                     if
                               f=l-(A-height)/L1;
                                                              f=1;
         else
(height>4.31835)
                                 }
                                                            source
                                                                               =
         {
                                else if (height>B)
                                                        MaxPower*f*fuel_map;
             f=0;
                                 {
                                                            return source;
                                                        #endif /* !PARRLLEL*/
        }
    else
                                f=1-(height-B)/L2;
                                                        }
       f=1;
                                 }
    source
                                   else
                                                if
                       =
MaxPower*f*fuel_map;
                           (height>4.31835)
                                                        DEFINE_SOURCE(suorce_GE8X8_
                                                        20Y, c, t, dS, eqn)
   return source;
                                     {
#if !PR_HOST
                                         f=0;
                                                        {
                                     }
                                                            int row, columes;
    if (A> height)
                                else
                                                            real x[ND_ND];
                                   f=1;
    {
                                                            real Fxz[6][6]=
                                source
                                                   =
                                                             {
                                                                 /*Ζ軸方向→*/
    f=l-(A-height)/L1;
                            MaxPower*f*fuel_map;
                                return source;
                                                             /*
    }
    else if (height>B)
                           #endif /* !RP_NODE*/
                                                        */308.546,308.546,308.546,3
    {
                                                        8.43,38.43,38.43,
                                                           /*
                                                                              Х
                                                        */308.546,308.546,308.546,3
    f=1-(height-B)/L2;
                          #if !PARALLEL
                                                        8.43,38.43,38.43,
    }
                                                           /*
         else if
                               if (A> height)
                                                                             軸
(height>4.31835)
                                {
                                                        */308.546,308.546,38.43,38.
                                                        43,38.43,0 ,
         {
                                                           /*
              f=0;
                                f=1-(A-height)/L1;
                                                                             方
                                                        */38.43,38.43,38.43,38.43,3
         }
                                 }
```

```
8.43,0 ,
                                  }
                                                          #endif /* !PR_HOST*/
   /*
                     向
                                 else if (height>B)
                                                          #if !RP_NODE
*/38.43,38.43,38.43,38.43,0
                                  {
                                                               if (A> height)
  ,0 ,
   /*↓*/38.43,38.43,0 ,
                               f=1-(height-B)/L2;
                                                               {
0 ,0 ,0 ,
                                  }
   };
                                                   if
                                                               f=1-(A-height)/L1;
                                      else
                             (height>4.31835)
                                                               }
    real A, B, L1, L2, f, p0,
                                      {
                                                               else if (height>B)
                                         f=0;
source, height, MaxPower,
                                                               {
fuel_w, pitch, fuel_map,
                                  }
                                 else
                                                               f=l-(height-B)/L2;
axis_x, axis_z;
    C_{CENTROID(x, c, t)};
                                  f=1;
                                                               }
    height= x[1];
                                 source
                                                                   else if
                                                     =
    p0=0.66045;
                            MaxPower*f*fuel_map;
                                                          (height>4.31835)
    MaxPower=1.0/0.0657400
                                 return source;
                                                                   {
8460044/0.825;
                            #if !PR_HOST
                                                                        f=0;
         L1 = 0.15*3.6579;
                                                                   }
         L2 = 3.6579*0.8;
                                if (A> height)
                                                               else
         A = p0+L1;
                                  {
                                                                  f=1;
         B = p0+L2;
                                                               source
                                                                                  =
                                                          MaxPower*f*fuel_map;
                                  f=1-(A-height)/L1;
    pitch=0.156624;
                                  }
                                                               return source:
    fue1_w=0.13406/2.0;
                                 else if (height>B)
                                                          #endif /* !RP_NODE*/
    axis_x=
                                  {
(x[0]+fuel_w)/pitch;
    axis_z=
                                 f=1-(height-B)/L2;
                                                        #if !PARALLEL
(x[2]+fuel_w)/pitch;
                                  }
    row=axis_z;
                                      else
                                                  i f
                                                             if (A> height)
    columes=axis_x;
                           (height>4.31835)
                                                               {
    fuel_map=Fxz[row][colu
                                      {
mes];
                                           f=0;
                                                               f=1-(A-height)/L1;
                                      }
                                                               }
                                  else
                                                               else if (height>B)
    if (A> height)
                                      f=1;
                                                               {
                                 source
    {
                                                     =
                             MaxPower*f*fuel_map;
                                                               f=1-(height-B)/L2;
    f=1-(A-height)/L1;
                                 return source;
                                                               }
```

```
92
```

```
else
                       if };
                                                                        else
                                                                                     i f
(height>4.31835)
                                                             (height>4.31835)
          {
                                   real A, B, L1, L2, f, p0,
                                                                        {
               f=0;
                              source, height, MaxPower,
                                                                             f=0;
                               fuel_w, pitch, fuel_map,
          }
                                                                        }
     else
                              axis_x, axis_z;
                                                                  else
         f=1;
                                   C\_CENTROID(x, c, t);
                                                                      f=1;
     source
                         =
                                   height = x[1];
                                                                   source
                                                                                       =
MaxPower*f*fuel_map;
                                    p0=0.66045;
                                                             MaxPower*f*fuel_map;
                                   MaxPower=1.0/0.0657400
                                                                   return source;
    return source;
#endif /* !PARRLLEL*/
                              8460044/0.825;
                                                             #if !PR HOST
                                        L1 = 0.15*3.6579;
}
                                        L2 = 3.6579*0.8;
                                                                   if (A> height)
DEFINE_SOURCE(suorce_GE8X8_
                                        A = p0+L1;
                                                                   {
40Y, c, t, dS, eqn)
                                         B = p0+L2;
                                                                   f=1-(A-height)/L1;
{
    int row, columes;
                                   pitch=0.156624;
                                                                   }
     real x[ND_ND];
                                    fuel_w=0.13406/2.0;
                                                                  else if (height>B)
     real Fxz[6][6]=
                                   axis_x=
                                                                   {
     {
                              (x[0]+fuel_w)/pitch;
          /*Z軸方向→*/
                                                                   f=1-(height-B)/L2;
                                    axis_z=
     /*
                              (x[2]+fuel_w)/pitch;
                                                                   }
*/191.166,191.166,191.166,6
                                    row=axis_z;
                                                                        else
                                                                                      if
2.036,62.036,62.036,
                                                             (height>4.31835)
                                    columes=axis_x;
    /*
                        Х
                                   fuel_map=Fxz[row][colu
                                                                        {
*/191.166,191.166,191.166,6
                                                                             f=0;
                              mes];
2.036,62.036,62.036,
                                                                        }
     /*
                        軸
                                                                   else
*/191.166,191.166,62.036,62
                                   if (A> height)
                                                                        f=1;
.036,62.036,0 ,
                                                                   source
                                    {
                                                                                       =
   /*
                                                             MaxPower*f*fuel_map;
                        方
*/62.036,62.036,62.036,62.0
                                    f=1-(A-height)/L1;
                                                                  return source;
36,62.036,0 ,
                                                             #endif /* !PR_HOST*/
                                    }
     /*
                        向
                                    else if (height>B)
                                                             #if !RP_NODE
*/62.036,62.036,62.036,62.0
                                    {
36,0 ,0 ,
                                                                   if (A> height)
    /*↓*/62.036,62.036,0
                                    f=1-(height-B)/L2;
                                                                   {
 ,0 ,0 ,0 ,
                                    }
```

```
93
```

```
f=1-(A-height)/L1;
                                        f=1;
                                                               axis_x, axis_z;
     }
                                                                    C\_CENTROID(x, c, t);
                                    source
                                                         =
     else if (height>B)
                               MaxPower*f*fuel_map;
                                                                    height = x[1];
                                                                    p0=0.66045;
     {
                                    return source;
                               #endif /* !PARRLLEL*/
                                                                    MaxPower=1.0/0.0657400
     f=1-(height-B)/L2;
                                                               8460044/0.825;
                               }
                                                                         L1 = 0.15*3.6579;
     }
          else
                        if
                                                                         L2 = 3.6579*0.8;
(height>4.31835)
                               DEFINE_SOURCE(suorce_GE8X8_
                                                                         A = p0+L1;
          {
                               80Y, c, t, dS, eqn)
                                                                         B = p0+L2;
               f=0;
                               {
          }
                                    int row, columes;
                                                                    pitch=0.156624;
     else
                                                                    fuel_w=0.13406/2.0;
                                     real x[ND_ND];
          f=1;
                                    real Fxz[6][6]=
                                                                    axis_x=
     source
                         =
                                     {
                                                               (x[0]+fuel_w)/pitch;
MaxPower*f*fuel_map;
                                          /*Z軸方向→*/
                                                                    axis_z=
                                    /*
    return source;
                                                               (x[2]+fuel_w)/pitch;
#endif /* !RP_NODE*/
                               */73.382,73.382,73.382,14.7
                                                                    row=axis_z;
                               55,14.755,14.755,
                                                                    columes=axis_x;
                                    /*
                                                        Х
                                                                    fuel_map=Fxz[row][colu
#if !PARALLEL
                               */73.382,73.382,73.382,14.7
                                                               mes]:
                               55,14.755,14.755,
                                    /*
     if (A> height)
                                                         軸
                               */73.382,73.382,14.755,14.7
                                                                    if (A> height)
     {
                               55,14.755,0 ,
                                                                    {
     f=1-(A-height)/L1;
                                   /*
                                                        方
                               */14.755,14.755,14.755,14.7
     }
                                                                    f=1-(A-height)/L1;
                               55,14.755,0 ,
     else if (height>B)
                                                                    }
                                     /*
                                                        向
                                                                    else if (height>B)
     {
                               */14.755,14.755,14.755,14.7
                                                                    {
     f=1-(height-B)/L2;
                               55,0 ,0 ,
                                    /*↓*/14.755,14.755,0
     }
                                                                    f=1-(height-B)/L2;
          else
                        if
                                ,0 ,0 ,0 ,
                                                                    }
(height>4.31835)
                                                                         else
                                                                                      if
                                    };
          {
                                                               (height>4.31835)
               f=0;
                                    real A, B, L1, L2, f, p0,
                                                                         {
                               source, height, MaxPower,
                                                                               f=0;
          }
                               fuel_w, pitch, fuel_map,
     else
                                                                         }
```

```
94
```

else f=1-(height-B)/L2; } f=1; } else if source = MaxPower*f*fuel_map; (height>4.31835) return source; { **#if** !PR HOST f=0; { } if (A> height) else f=1; { source = { f=1-(A-height)/L1; MaxPower*f*fuel_map; } /* return source; #endif /* !RP_NODE*/ else if (height>B) { /* f=1-(height-B)/L2;**#if** !PARALLEL } else i f if (A> height) /* (height>4.31835) { { f=0; f=1-(A-height)/L1; } } else else if (height>B) f=1; /* { source = MaxPower*f*fuel_map; f=1-(height-B)/L2;return source; } #endif /* !PR_HOST*/ else if #if !RP_NODE (height>4.31835) }; { f=0: if (A> height) { } else f=1-(A-height)/L1; f=1; } source = else if (height>B) MaxPower*f*fuel_map; height= x[1]; { return source; #endif /* !PARRLLEL*/

DEFINE_SOURCE(suorce_GE8X8_ 100Y, c, t, dS, eqn) int row, columes; real x[ND_ND]; real Fxz[6][6]= /*Z軸方向→*/ */45.468,45.468,45.468,9.14 19,9,1419,9,1419, Х */45.468,45.468,45.468,9.14 19,9.1419,9.1419, 軸 */45.468,45.468,9.1419,9.14 19,9.1419,0 , /* 方 */9.1419,9.1419,9.1419,9.14 19,9.1419,0 , 向 */9.1419,9.1419,9.1419,9.14 19,0 ,0 , /*↓*/9.1419,9.1419,0 ,0 ,0 ,0 , real A, B, L1, L2, f, p0, source, height, MaxPower, fuel_w, pitch, fuel_map, axis_x, axis_z; $C_CENTROID(x, c, t);$

> p0=0.66045; MaxPower=1.0/0.0657400

8460044/0.825;	#if !PR_HOST	f=0;								
L1 = 0.15*3.6579;		}								
L2 = 3.6579*0.8;	if (A> height)	else								
A = p0+L1;	{	f=1;								
B = p0+L2;		source =								
	f=1-(A-height)/L1;	<pre>MaxPower*f*fuel_map;</pre>								
pitch=0.156624;	}	return source;								
fuel_w=0.13406/2.0;	else if (height>B)	<pre>#endif /* !RP_NODE*/</pre>								
axis_x=	{									
(x[0]+fuel_w)/pitch;										
axis_z=	f=1-(height-B)/L2;	#if !PARALLEL								
(x[2]+fuel_w)/pitch;	}									
row=axis_z;	else if	if (A> height)								
columes=axis_x;	(height>4.31835)	{								
fuel_map=Fxz[row][colu	{									
mes];	f=0;	f=1-(A-height)/L1;								
	}	}								
	else	else if (height>B)								
if (A> height)	f=1;	{								
{	source =									
	MaxPower*f*fuel_map;	f=1-(height-B)/L2;								
f=1-(A-height)/L1;	return source;	}								
}	<pre>#endif /* !PR_HOST*/</pre>	else if								
else if (height>B)	#if !RP_NODE	(height>4.31835)								
{		{								
	if (A> height)	f=0;								
f=1-(height-B)/L2;	{	}								
}		else								
else if	f=1-(A-height)/L1;	f=1;								
(height>4.31835)	}	source =								
{	else if (height>B)	<pre>MaxPower*f*fuel_map;</pre>								
f=0;	{	return source;								
}		<pre>#endif /* !PARRLLEL*/</pre>								
else	f=1-(height-B)/L2;	}								
f=1;	}									
source =	else if									
<pre>MaxPower*f*fuel_map;</pre>	(height>4.31835)									
return source;	{									

```
DEFINE_SOURCE(suorce_ATRIUM
_10, c, t, dS, eqn)
{
    int row, columes;
    real x[ND_ND];
    real Fxz[6][6]=
     {
          /*Z軸方向→*/
    /*
*/500.,500.,500.,100.,100.,
100.,
    /*
                        Х
*/500.,500.,500.,100.,100.,
100.,
   /*
                        軸
*/500.,500.,100.,100.,100.,
0,
    /*
                        方
*/100.,100.,100.,100.,100.,
0,
   /*
                        向
*/100.,100.,100.,100.,0
                         .
0,
    /*↓*/100.,100.,0
                        ,0
  ,0 ,0 ,
    };
    real A, B, f, p0, source,
height, Maxpower, act_length,
fuel_w, pitch, fuel_map,
axis_x, axis_z;
    C\_CENTROID(x, c, t);
     height= x[1];
     p0=0.66045;
```

act_length=3.6579;

th:

A=(height-p0)/act_leng

Maxpower=1.0/(0.017972)

```
*act_length);
    pitch=0.156624;
    fuel_w=0.13406/2.0;
    axis_x=
(x[0]+fuel_w)/pitch;
    axis_z=
(x[2]+fuel_w)/pitch;
    row=axis_z;
    columes=axis_x;
fuel_map=Fxz[row][columes];
    if (p0< height &&</pre>
```

```
height < (p0+act_length) )
{
    f=-61.266*pow(A,
6)+168.83*pow(A,
5)-187.298*pow(A,
4)+109.4*pow(A,
3)-38.146*pow(A,
2)+8.1456*A+0.3738;
}
else</pre>
```

```
f=0;
```

```
source
                          =
Maxpower*f*fuel_map;
     return source;
#if !PR_HOST
                i f
                       (p0<
       &&
              height
height
                         <
(p0+act_length) )
    {
          f = -61.266 * pow(A)
6)+168.83*pow(A,
5)-187.298*pow(A,
4)+109.4*pow(A)
3)-38.146*pow(A,
```

```
2)+8.1456*A+0.3738;
}
else
f=0;
```

```
source
Maxpower*f*fuel_map;
return source;
#endif /* !PR_HOST*/
```

#if !RP_NODE

=

```
if (p0< height &&
height < (p0+act_length) )
        {
            f=-61.266*pow(A,
6)+168.83*pow(A,
5)-187.298*pow(A,
4)+109.4*pow(A,
3)-38.146*pow(A,
3)-38.146*pow(A,
2)+8.1456*A+0.3738;
        }
        else</pre>
```

```
f=0;
source =
Maxpower*f*fuel_map;
return source;
#endif /* !RP_NODE*/
```

#i f ! PARALLEL

```
if (p0< height &&
height < (p0+act_length) )
{
    f=-61.266*pow(A,</pre>
```

```
6)+168.83*pow(A,
5)-187.298*pow(A,
4)+109.4*pow(A,
3)-38.146*pow(A,
2)+8.1456*A+0.3738;
}
else
```

```
f=0;
```

source =
Maxpower*f*fuel_map;
return source;
#endif /* !PARRLLEL*/
}

附錄二

Y	AMBIENT TEMP	Y	CGR		1 2		3 4	5	10) 15	20	25	30	35	40	45	5 50	55	60	65	70	75	80	85	90	95
0.002286	32 °C	0.002286	0.39517	8 0.386118	8 0.377451	0.36915	5 0.361214	0.353608	0.320073	3 0.292802	0.270454	0.252005	0.236662	0.223809	0.212962	0.203741	0.195843	0.18903	0.183113	0.177946	0.173415	0.169439	0.165961	0.162946	0.160381	0.158271
0.050752	AMDIENT DU	0.050752		0 0	0 0		0.413709	0.403952	0.361194	0.32677	0.298815	0.275926	0.25703	0.241306	0.228115	0.216961	0.207454	0.199288	0.192225	0.186077	0.187314	0.1/6004	0.171902	0.158354	0.160801	0.166996
0.147682	80 %	0.147682		0 0	0 0			0.400001	0.431551	0.384211	0.346268	0.315567	0.290493	0.269829	0.252645	0.238229	9 0.226027	0.215614	0.206658	0.198904	0.192161	0.186285	0.181177	0.176775	0.173049	0.169998
0.196147	AMBIENTe	0.196147		0 0	0 0		0 0	0 0	0.455834	0.403864	0.362375	0.328926	0.301697	0.279323	0.260767	0.245236	5 0.23212	0.220949	0.211358	0.203068	0.195868	0.189603	0.184164	0.179481	0.175521	0.172281
0.244613	35.44711	0.244613		0 0	0 0) (0 0	(0.420726	0.376148	0.340313	0.31122	0.287372	0.267636	0.251151	0.237254	0.225436	0.215305	0.20656	0.198973	0.192379	0.18666	0.181741	0.177583	0.174185
0.293078	P0	0.293078		0 0	0 0		0 (0 0	(0.435992	0.388581	0.350566	0.319774	0.294587	0.273782	0.256434	0.241832	0.229432	0.218815	0.209661	0.201729	0.19484	0.188871	0.18374	0.179407	0.175867
0.390009	0.000201	0.390009		0 0	0 0		0 0	0 0	(0.450511	0.41142	0.369337	0.335388	0.30772	0.284942	0.266005	5 0.240008	0.236642	0.225138	0.212521	0.206678	0.199256	0.190904	0.18732	0.18267	0.178875
0.438474	QG	0.438474		0 0	0 0		0 0	0 0	(0 0	0.422797	0.378659	0.34312	0.314207	0.290441	0.270711	0.254171	0.240176	0.228232	0.217966	0.209093	0.201408	0.194762	0.189061	0.184256	0.180335
0.486939	40000 J/mol	0.486939		0 (0 0) (0 0	() 0	0.433358	0.387296	0.350272	0.320197	0.295513	0.275046	5 0.257908	0.243422	0.231073	0.220466	0.211307	0.203378	0.196527	0.190654	0.185706	0.18167
0.535404	R	0.535404		0 0	0 0	1	0 (0 0	(0 0	0.444625	0.396493	0.357875	0.326556	0.300888	0.279635	5 0.26186	0.246852	0.23407	0.223102	0.213639	0.205453	0.198385	0.192329	0.187229	0.183073
0.58387	8.314 J/mol-K	0.58387		0 0	0 0				() 0	0.456198	0.405923	0.365657	0.333055	0.306374	0.284312	2 0.265883	0.25034	0.237116	0.225779	0.216005	0.207557	0.200267	0.194025	0.188771	0.184491
0.6808	353.15	0.6808		0 0	0 0		o d	0 0	(0	0	0.425345	0.381644	0.346374	0.317595	0.293861	0.274082	0.257438	0.243305	0.23121	0.2208	0.211816	0.202170	0.197452	0.191886	0.187355
0.729266	α	0.729266		0 0	0 0		0 (0 0	() 0	0	0.434927	0.389512	0.352915	0.323094	0.298531	0.278086	0.260899	0.246317	0.23385	0.223129	0.213882	0.205919	0.199112	0.193393	0.188739
0.777731	2.2	0.777731		0 0	0 0		0 (0 0	(0 0	0	0.444192	0.397108	0.35922	0.328388	0.303023	3 0.281933	0.26422	0.249207	0.236381	0.225359	0.215859	0.207683	0.200699	0.194833	0.190062
0.826196	THRESHOLD 20	0.826196		0 0	0 0		0 (0 0	(0 0	0	0.453285	0.404552	0.365391	0.333564	0.307408	8 0.285684	0.267457	0.25202	0.238843	0.227527	0.21778	0.209397	0.202239	0.19623	0.191345
0.074001	I(C) 32 RH% 25%	0.074001		0 0	0 0			0 0	() 0	0	0.402541	0.411955	0.37756	0.350097	0.311733	3 0.2093048	0.270639	0.257528	0.241274	0.229007	0.219075	0.212739	0.205736	0.197606	0.192607
0.971592		0.971592		0 0	0 0		0 (0 0	(0	0	0	0.426478	0.383521	0.348733	0.320237	0.296638	0.27689	0.260207	0.245997	0.233819	0.223349	0.214358	0.206693	0.200267	0.195049
1.020057	loading power	1.020057		0 0	0 0) (0 0	() 0	0	0	0.433682	0.389464	0.353694	0.324424	4 0.300207	0.279959	0.262866	0.248317	0.235857	0.225151	0.215962	0.208132	0.20157	0.196243
1.068523	17 kw	1.068523		0 0	0 0		0 (0 0	(0 0	0	0	0.440982	0.39548	0.358711	0.328654	1 0.303809 7 0.307491	0.283053	0.265545	0.250654	0.237908	0.226963	0.217574	0.209577	0.202878	0.197443
1.1165453	大氣佰比 50%	1.1165453			0 0						0	0	0.448455	0.40163	0.363834	0.3329	0.307481	0.286205	0.268272	0.25303	0.239993	0.228804	0.219211	0.212532	0.204206	0.198659
1.213917		1.213917		0 0	0 0		0 0	0 0	() 0	0	0	0.450057	0.41426	0.374339	0.341806	5 0.314988	0.292642	0.273835	0.257873	0.244238	0.23255	0.222539	0.214025	0.206902	0.201128
1.262381		1.262381		0 (0 0) (0 0	() 0	0	0	0	0.420603	0.379607	0.34623	3 0.318743	0.295857	0.276611	0.260288	0.246352	0.234413	0.224193	0.215505	0.20824	0.202353
1.310846		1.310846		0 0	0 0		0 (0 0	(0 0	0	0	0	0.42698	0.384897	0.350669	9 0.322506	0.299076	0.279389	0.262702	0.248465	0.236275	0.225845	0.216983	0.209575	0.203575
1.35931		1.35931		0 0	0 0) () (0 0	(0	0	0	0.433367	0.39019	0.350500	0 320205 0 32000	0.302291	0.28216	0.267489	0.25057	0.238129	0.22/489	0.210905	0.210903	0.205088
1.456238		1.456238		0 0	0 0 0		o c	0 0	(0 0	0	0	0	0.446068	0.4007	0.363905	5 0.333709	0.308648	0.287635	0.26986	0.254722	0.241782	0.230727	0.221347	0.213515	0.207178
1.504703		1.504703		0 0	0 0) (0 0	() 0	0	0	0	0.452511	0.406023	0.368356	5 0.33747	0.311856	0.290395	0.272253	0.256811	0.243619	0.232354	0.2228	0.214826	0.208376
1.553167		1.553167		0 0	0 0		0 (0 0	() 0	0	0	0	0.459163	0.411514	0.372943	3 0.341342	0.315157	0.293233	0.274712	0.258957	0.245505	0.234024	0.224291	0.21617	0.209604
1.601631		1.601631		0 0	0 0) (0 0	(0 0	0	0	0	0	0.417091	0.377597	7 0.345269	0.318502	0.296107	0.2772	0.261127	0.247411	0.23571	0.225795	0.217527	0.210843
1.698559		1.698559		0 0	0 0		0 0	0 0	() 0	0	0	0	0	0.42849	0.387097	0.349321	0.325312	0.299000	0.219702	0.265533	0.249371	0.239129	0.228844	0.220274	0.212110
1.747024		1.747024		0 0	0 0		0 (0 0	() 0	0	0	0	0	0.434113	0.391778	0.357212	0.32866	0.304822	0.284736	0.267692	0.253171	0.240802	0.230335	0.221617	0.214576
1.795488		1.795488		0 0	0 0) (0 0	() 0	0	0	0	0	0.439805	0.396512	2 0.361193	0.33204	0.307719	0.287238	0.26987	0.255079	0.242488	0.231837	0.222968	0.215809
1.843952		1.843952		0 0	0 0	1	0 (0 0	(0 0	0	0	0	0	0.44557	0.401302	2 0.365217	0.335456	0.310643	0.289763	0.272065	0.257003	0.244185	0.233349	0.224329	0.21705
1.892416		1.892416		0 0	0 0				(0	0	0	0	0.451427	0.406164	1 0.3693 0 373425	0.338918	0.313605	0.292318	0.274286	0.258947	0.245901	0.234876	0.225703	0.218302
1.989345		1.989345		0 0	0 0		0 0) 0	() 0	0	0	0	0	0,457550	0.416059	0.377597	0.345946	0.319613	0.294090	0.278783	0.262882	0.249371	0.237962	0.228477	0.220831
2.037809		2.037809		0 0	0 0) (0 0	() 0	0	0	0	0	0	0.421096	5 0.381816	0.349516	0.322662	0.300122	0.281061	0.264874	0.251126	0.239522	0.229879	0.222108
2.086273		2.086273		0 (0 0		0 (0 0	(0 0	0	0	0	0	0	0.426203	3 0.38609	0.353131	0.325747	0.302776	0.283363	0.266885	0.252898	0.241097	0.231293	0.223395
2.134738		2.134738		0 0	0 0			0 0	(0 0	0	0	0	0	0	0.431391	0.390428	0.356796	0.328873	0.305465	0.285694	0.26892	0.254689	0.242688	0.232722	0.224696
2.231666		2.231666		0 0	0 0			0 0	() 0	0	0	0	0	0	0.430001	0.399297	0.364282	0.335251	0.310945	0.29044	0.273062	0.258333	0.244297	0.234107	0.220011
2.28013		2.28013		0 0	0 0		0 0	0 0	() 0	0	0	0	0	0	0.447402	2 0.403797	0.368076	0.33848	0.313717	0.292838	0.275154	0.260172	0.247554	0.237088	0.228668
2.328593		2.328593		0 0	0 0		0 (0 0	() 0	0	0	0	0	0	0.45292	2 0.408396	0.371951	0.341776	0.316545	0.295284	0.277286	0.262045	0.249215	0.238578	0.230022
2.377056		2.377056		0 0	0 0) (0 0	(0 0	0	0	0	0	0	(0.413145	0.375949	0.345175	0.319459	0.297803	0.27948	0.263972	0.250923	0.240109	0.231414
2.42002		2.42002		0 0	0 0			0	() 0	0	0	0	0	0		0.417990	0.384208	0.340041	0.325466	0.300308	0.283995	0.200955	0.252001	0.241005	0.232020
2.522446		2.522446		0 0	0 0		0 (0 0	() 0	0	0	0	0	0	(0.428067	0.388493	0.355823	0.328577	0.305675	0.28633	0.269982	0.256246	0.244876	0.235744
2.570909		2.570909		0 0	0 0		0 (0 0	() 0	0	0	0	0	0	(0.433307	0.392892	0.359551	0.331765	0.308424	0.28872	0.272077	0.2581	0.246535	0.23725
2.619372		2.619372		0 0	0 0		0 (0 0	(0 0	0	0	0	0	0	0	0.438691	0.397407	0.363376	0.335034	0.311241	0.291167	0.274222	0.259996	0.248231	0.238789
2.667836		2.667836		0 0	0 0				(0	0	0	0	0	(0.444565	0.40233	0.367542	0.338593	0.314306	0.293828	0.276552	0.262056	0.250074	0.24046
2.764762		2.764762		0 0	0 0		0 0	0 0	() 0	0	0	0	Ő	0	(0.455887	0.411806	0.375554	0.345428	0.320187	0.298931	0.281016	0.266	0.253598	0.243656
2.813225		2.813225		0 0	0 0) (0 0	() 0	0	0	0	0	0	(0 0	0.416615	0.379616	0.34889	0.323163	0.30151	0.283271	0.267991	0.255376	0.245267
2.861688		2.861688		0 0	0 0	1	0 (0 0	(0 0	0	0	0	0	0	(0 0	0.421466	0.38371	0.352377	0.326159	0.304105	0.285539	0.269992	0.257163	0.246885
2.910152		2.910152		0 0	0 0				(0	0	0	0				0.426349	0.387828	0.355882	0.329168	0.306711	0.287815	0.272	0.258954	0.248507
3.007078		3.007078		0 0	0 0		0 0	0 0	(0	0	0	0	ő	0		0 0	0.436203	0.396129	0.362941	0.335223	0.311949	0.292387	0.276029	0.262547	0.250135
3.055541		3.055541		0 0	0 0) (0 0	() 0	0	0	0	0	0	(0 0	0.441135	0.400279	0.366467	0.338245	0.314561	0.294664	0.278035	0.264335	0.253377
3.104004		3.104004		0 0	0 0		0 (0 0	(0 0	0	0	0	0	0	(0 0	0.446107	0.40446	0.370017	0.341285	0.317188	0.296954	0.280051	0.266131	0.255001
3.152468		3.152468		0 0	0 0			0 0	() 0	0	0	0	0	0	(0 0	0.451123	0.408675	0.373593	0.344347	0.319831	0.299257	0.282077	0.267936	0.256633
3.249394		3.249394		0 0	0 0			0 0	() 0	0	0	0	0	0		0 0	0	0.412943	0.380896	0.350593	0.325219	0.303948	0.286203	0.209738	0.259951
3.297857		3.297857		0 0	0 0		0 0	0 0	(0 0	0	0	0	0	0	(0 0	0	0.421752	0.384673	0.35382	0.328002	0.306368	0.28833	0.2735	0.26166
3.34632		3.34632		0 0	0 0	1	0 0	0 0	(0 0	0	0	0	0	0	(0 0	0	0.426379	0.388589	0.357164	0.330883	0.308873	0.29053	0.275457	0.263427
3.394783		3.394783		0 0	0 0			0	(0	0	0	0	0	0	(0 0	0	0.431181	0.39265	0.36063	0.333867	0.311466	0.292807	0.277481	0.265253
3.491708		3.491708		0 0	0 0) (, U) 0	() 0	0	0	0	0	0	(, U) 0	0	0.441322	0.401216	0.367934	0.34015	0.316923	0.297595	0.281734	0.269089
3.54017		3.54017		0 0	0 0	1	0 0	0 0	(0 0	0	0	0	0	0	() Ő	0	0.446733	0.405783	0.371823	0.343494	0.319823	0.300138	0.283991	0.271124
3.588632		3.588632		0 0	0 0	1	0 (0 0	(0 0	0	0	0	0	0	(0 0	0	0.452411	0.410571	0.375899	0.346994	0.322859	0.302798	0.286351	0.273251
3.637094		3.637094		0 0	0 0			0 0	(0 0	0	0	0	0	0	(0 0	0	0	0.415562	0.380144	0.350638	0.326017	0.305563	0.288804	0.275461
3,734018		3,734018		0 0	0 0) 0	() 0	0	0	0	0	0	. () 0) 0	0	0	0.425258	0,388382	0.357703	0.332133	0,310916	0.293547	0.279731
3.78248		3.78248		0 0	0 0		o c	00	(00	0	0	0	0	0	() Ő	0	0	0.430343	0.392698	0.361401	0.335332	0.313713	0.296024	0.28196
3.830942		3.830942		0 (0 0) (0 0	() 0	0	0	0	0	0	(0 0	0	0	0.43541	0.396996	0.36508	0.338513	0.316493	0.298485	0.284173
3.879405		3.879405		0 0	0 0	1	0 (0 0	(0 0	0	0	0	0	0	(0 0	0	0	0.440499	0.40131	0.368771	0.341702	0.319278	0.30095	0.286389
3.927867		3.92/867		0 0	0 0) () (0 0	(0	0	0	0	0	((0 0	0	0	0.445689	0.405706	0.376422	0.348307	0.325043	0.303456	0.288641
4.024791		4.024791		0 0	0 0		ő (0	(0	0	0	0	Ő	0	0) Ö	0	0	0.451071	0.415026	0.380491	0.351817	0.328103	0.308751	0.293397
4.073253		4.073253		0 0	0 0		0 (0 0	(0 0	0	0	0	0	0	(0 0	0	0	0	0.420355	0.38504	0.355737	0.33152	0.311768	0.296105
4.121715		4.121715		0 0	0 0	1	0 0	0 0	(0 0	0	0	0	0	0	(0 0	0	0	0	0.425797	0.38968	0.359733	0.335001	0.31484	0.298862
4.170177		4.1/0177		0 0	0 0			0	(0 0	0	0	0	0	0	(0 0	0	0	0	0.43184	0.394828	0.364165	0.338858	0.318243	0.301913
4.267102		4,267102		0 0	0 0			, U) 0	() 0	0	0	0	0	0	((0 0	0	0	0	0.443908	0.405099	0.372995	0.346537	0,32501	0.307976
4.315564		4.315564		0 0	0 0		o c	0 0		0 0	0	Ő	0	0	0	(0 0	0	0	0	0.44972	0.410039	0.377237	0.350222	0.328255	0.310882
4.364026		4.364026		0 0	0 0	1	0 (0 0	(0 0	0	0	0	0	0	(0 0	0	0	0	0	0.414166	0.38078	0.353298	0.330962	0.313305
4.412488		4.412488		0 0	0 0		0 0	0 0	(0 0	0	0	0	0	0	(0 0	0	0	0	0.451667	0.415797	0.382179	0.354513	0.332031	0.314261
4.40095		4.40095		0 0	0 0			, 0	(0	0	0	0	0	0	0 0	0	0	0	0.437868	0.39996	0.368579	0.331456	0.329341	0.304946
4.557873		4.557873		0 0	0 0		0 0	0 0	(0 0	0	- 0	0	0	0	(0 0	0	0	0	0.421136	0.385705	0.356311	0.33202	0.312209	0.296501
4.606334		4.606334		0 0	0 0	1	0 0	0 0	(0 0	0	0	0	0	0	(0 0	0	0	0.451229	0.410395	0.376537	0.348406	0.325129	0.306123	0.291037
4.654795		4.654795		0 0	0 0		0 0	0 0	(0 0	0	0	0	0	0	0	0 0	0	0	0.440354	0.401187	0.368666	0.341611	0.319199	0.30088	0.286326
4.703256		4.703256		0 0	0 0			0 0	(0	0	0	0	0	((0 0	0	0	0.450232	0.392604	0.356665	0.331235	0.313652	0.295971	0.281912
4.800179		4.800179		0 0	0 0		0 0	0 0		0 0	0	0	0	0	0	0	0 0	0	0	0.417725	0.381983	0.352216	0.327383	0.306759	0.289864	0.276416
附件三		.54364,0.550201,0.5566		9266,0.847936}																						
-----	------------------------	-------------------------	---	------------------------------																						
		75,0.56358,0.569648,0.	•	<pre>begin_f_loop(f,t)</pre>																						
•		576169,0.582696,0.589	•	{																						
•	#include"udf.h"	233,0.595781,0.602352,	•	C_CENTROID(f,t,i)																						
•	DEFINE_PROFILE(TS	0.608956,0.65596,0.622		;																						
	C_profile,t,i)	264,0.628919,0.635655,	•	Y_AXIS=x[1];																						
•	{	0.642542,0.649505,0.65	•																							
•	real x[ND_ND],	6568,0.663745,0.6,104,		F_PROFILE(f,t,i)=1																						
	Y_AXIS;	0.678456,0.686458,0.69		7* T_FACTOR;																						
•	face_t f;	4032,0.701628,0.70921	•	}																						
•	real T_FACTOR[100]=	1,0.716785,0.724333,0.,	•	end_f_loop(f,t)																						
	{0.219168,0.250364,0.2	31858,0.73934,0.74674	•																							
	77713,0.297241,0.3118	1,0.754131,0.761515,0.	•	}																						
	92,0.323978,0.334566,0	768924,0.776405,0.784																								
	.34421,0.353275,0.3622	009,0.91825,0.799859,0																								
	62,0.37042,0.378936,0.	.808092,0.816575,0.825																								

100

36,0.83448,0.843889,0.

853422,0.8,1892,0.8711

93,0.880367,0.88949,0.

898699,0.908151,0.917

944,0.928782,0.,39725,

0.951738,0.963605,0.97

5301,0.986452,0.99568,

0.999305,0.99016,0.9,3

577,0.930359,0.908429,

0.889231,0.870992,0.85

387492,0.396099,0.404

588,0.412774,0.420544,

0.428035, 0.435368, 0.44

2492,0.449427,0.4562,9

,0.463064,0.469939,0.4

76867,0.483777,0.4905

91,0.497351,0.504037,0

.51097,0.517081,0.5235

75,0.530198,0.536846,0